

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 03-211656

(43)Date of publication of application : 17.09.1991

(51)Int.Cl. G06F 15/16
G06F 12/06

(21)Application number : 01-338040

(71)Applicant : TEXAS INSTR INC <TI>

(22)Date of filing : 26.12.1989

(72)Inventor : KEITH BALMER
NICHOLAS K ING-SIMMONS
GUTTAG KARL M
GOVE ROBERT J

(30)Priority

Priority number : 89 435591

Priority date : 17.11.1989

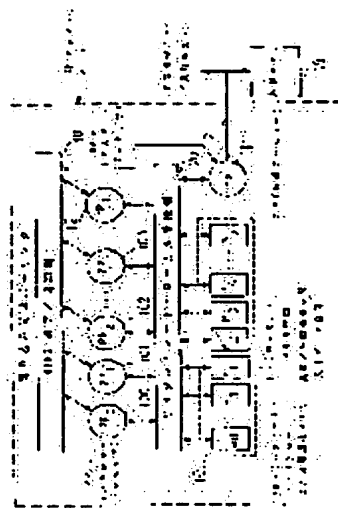
Priority country : US

(54) MULTIPROCESSOR SYSTEM USING CROSSBAR LINK OF PROCESSOR AND MEMORY AND OPERATING METHOD THEREFOR

(57)Abstract:

PURPOSE: To mutually connect all processors to any memory by all constitution by providing a multi-link multi-bus crossbar switch between the respective processors and the respective memories.

CONSTITUTION: This system is provided with the processors 100-103 capable of controlling many different processes by an instruction set supplied from the memory 10, the memory 10, a switch matrix 20 connected to the memory 10 and connected to the processors 100-103 and further, a master processor 12 and an MIMD(multiplex instruction multiplex data mode) communication/ synchronization network 40 for selectively and simultaneously enabling the switch matrix 20 by a processor cycle unit base, mutually connecting one processor and one memory and transferring the instruction set from a memory space and data from the other memory space. Thus, the different processors simultaneously and parallelly execute different tasks.



(19) 日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11) 特許番号

特許第3026984号
(P3026984)

(45) 発行日 平成12年3月27日 (2000.3.27)

(24) 登録日 平成12年1月28日 (2000.1.28)

(51) Int.Cl.

G 0 6 F 15/173
12/06

識別記号

5 4 0

F I

G 0 6 F 15/173
12/06

F

5 4 0 A

請求項の数 6 (全 85 頁)

(21) 出願番号

特願平1-338040

(22) 出願日

平成1年12月26日 (1989.12.26)

(65) 公開番号

特開平3-211656

(43) 公開日

平成3年9月17日 (1991.9.17)

審査請求日

平成8年12月25日 (1996.12.25)

(31) 優先権主張番号

4 3 5 5 9 1

(32) 優先日

平成1年11月17日 (1989.11.17)

(33) 優先権主張国

米国 (U S)

(73) 特許権者 999999999

テキサス インストルメンツ インコー
ポレイテッド
アメリカ合衆国 テキサス州 ダラス
ノース セントラル エクスプレスウェ
イ 13500

(72) 発明者

キース パルマー

英国 MK40 38A ベッドフォード
ソールカム クローズ 6

(72) 発明者

ニコラス ケイ イングシモンズ

英国 MK43 7SS オークリー ベ
ッドフォード リンクロフト 74

(74) 代理人

999999999

弁理士 中村 稔 (外7名)

審査官 石井 茂和

最終頁に続く

(54) 【発明の名称】 マルチプロセッサシステム

1

(57) 【特許請求の範囲】

【請求項1】 マルチプロセッサシステムにおいて、
複数のn個のプロセッサと、複数のm個のメモリと、ス
イッチマトリックスとを備え、
前記メモリの各々は、前記複数のm個のメモリ全てを含
む統一メモリ空間内の固有のアドレスの集合を有してお
り、ここでmはnよりも大であり、前記プロセッサの各
々は、前記統一メモリ空間内のアドレスを生成し、そし
て、多数の異なるプロセスを制御するために前記メモリ
の少なくとも1個から与えられる命令セットによって動
作可能であり、前記m個のメモリの各々の固有のアドレ
スの集合は、前記m個のメモリの他の前記固有のアドレ
スの集合とは別個で異なっており、前記m個のメモリの
各々の前記固有のアドレス指定可能なメモリ空間は、前
記n個のプロセッサの前記所定範囲のアドレス内であ

2

り、
前記スイッチマトリックスは前記メモリ及び前記プロセ
ッサに接続され、スイッチマトリックスを介してアドレ
ス及びデータの両方を前記プロセッサから前記メモリに
選択的に送るようになっており、第1のリンクのセット
と、第2のリンクのセットと、複数の又点とを含み、
前記第1のリンクの各々是对応するメモリに接続され、
前記第2のリンクの各々是对応するプロセッサに接続さ
れ、
前記複数の又点は前記第1のリンクと前記第2のリンク
との間の交点に配置され、又点の各々は、前記対応する
プロセッサによって生成されたアドレスに対して前記第
2のリンクを介して応答して、個々にサイクル単位ベー
スに動作して、前記第1のリンク及び前記第2のリンク
を接続し、それによって、もし、前記アドレスが、前記

対応するメモリに割り当てられた前記固有のアドレスの集合の範囲内にあるならば、プロセッサにそのメモリへのアクセスを認め、1個よりも多数のプロセッサが、前記対応するメモリの固有のアドレスの集合の範囲内のアドレスを生成する場合には、第1のリンクの各々に接続された前記又点は共に通信して、優先順位を決定し、それによって、最後に前記メモリへのアクセスが成功したプロセッサが一番低い優先順位を割り当てられ、それによって、前記n個のプロセッサの各々は、サイクルの各々で、前記m個のメモリ内の異なるメモリに同時にアクセス可能であるマルチプロセッサシステム。

【請求項2】前記n個のプロセッサの各々は、第1のデータポート及び第2のデータポートを含み、前記複数のm個のメモリは、前記n個のプロセッサの各々に対応するメモリの所定のセットから構成され、前記n個のプロセッサの各々の前記第1のデータポートに接続された前記第2のリンクは、前記m個のメモリの各々に接続された又点に接続され、それによって、前記n個のプロセッサの各々の前記第1のデータポートは前記m個のメモリの各々にアクセス可能であり、前記n個のプロセッサの各々の前記第2のデータポートに接続された前記第2のリンクは、そのプロセッサに対応するメモリの前記所定のセットに接続された又点にのみ接続され、それによって、前記n個のプロセッサの各々の前記第2のデータポートは、対応するメモリの前記所定のセットにのみアクセス可能である請求項1に記載のマルチプロセッサシステム。

【請求項3】前記n個のプロセッサの各々は命令ポートを更に含み、前記複数のm個のメモリは、前記n個のプロセッサの各々に対応するメモリの所定のセットから構成され、メモリの前記所定のセットの各々は、前記対応するプロセッサのための命令メモリを含み、SIMD/MIMDモードレジスタは前記スイッチマトリックスに接続され、単一命令複数データ (SIMD) モードもしくは複数命令複数データ (MIMD) モードのどちらかの指示をそこに記憶し、前記スイッチマトリックスは、前記プロセッサの数と等しい数の複数の隣接する部分を有する第3のリンクと、前記第3のリンクの隣接する部分間に配置された複数のバッファとを更に含み、前記第3のリンクの各々の部分は、対応するプロセッサの前記命令ポートに接続されており、前記複数のバッファは、単一命令複数データ (SIMD) モードの場合には、前記第3のリンクの前記隣接する部分を接続し、複数命令複数データ (MIMD) モードの場合には、前記第3のリンクの前記隣接する部分を隔離し、前記複数の又点は、(1) 第1のプロセッサの前記命令ポートに接続された前記第3のリンクの前記部分と前記第1のプロセッサに対応する第1の命令メモリに接続さ

れた第1のリンクとの交点に配置された第1の又点と、(2) 前記第1のプロセッサ以外のプロセッサの前記命令ポートに接続された前記第3のリンクの前記部分と前記対応する命令メモリに接続された前記第1のリンクとの交点に配置された第2の又点のセットと、(3) 前記第1のプロセッサに対応する前記命令メモリ以外の前記命令メモリに接続された前記第1のリンクと前記複数のプロセッサの前記データポートに接続された前記第2のリンクとの交点に配置された第3の又点のセットとを更に含み、

前記第1の又点は、接続を可能にするように常にイネーブルにされ、

前記第2の又点は、単一命令複数データ (SIMD) モードで動作する場合には、接続を禁止するようにディスエーブルにされ、複数命令複数データ (MIMD) モードで動作する場合には、接続を可能にするようにイネーブルにされ、

前記第3の又点のセットは、複数命令複数データ (MIMD) モードで動作する場合には、接続を禁止するようにディスエーブルにされ、単一命令複数データ (SIMD) モードで動作する場合には、接続を可能にするようにイネーブルにされ、

それによって、前記単一命令複数データ (SIMD) モードでは、前記プロセッサの各々の前記命令ポートは前記第1の命令メモリに接続され、前記複数命令複数データ (MIMD) モードでは、前記プロセッサの各々の前記命令ポートは前記対応する命令メモリに接続される請求項1に記載のマルチプロセッサシステム。

【請求項4】前記マルチプロセッサシステムは、前記マルチプロセッサシステムに対して外部にある装置への接続に適合する外部アドレス/データポートと、前記m個のメモリにアクセスするための前記スイッチマトリックスに接続され、また、前記外部アドレス/データポートに接続された転送プロセッサとを更に備えており、

前記転送プロセッサは、前記n個のプロセッサによってアクセスされていない前記m個のメモリの内の1つに、前記n個のプロセッサによるアクセスを妨害することなく同時にアクセス可能であり、前記転送プロセッサは、前記スイッチマトリックス及び前記外部アドレス/データポートを介して、前記m個のメモリと前記マルチプロセッサシステムに対して外部にある装置との間でデータを伝送し、前記マルチプロセッサシステムに対して外部にある前記装置は前記スイッチマトリックスに直接には接続されない請求項3に記載のマルチプロセッサシステム。

【請求項5】各々のプロセッサは、次の命令のアドレスを記憶しているプログラムカウンタと、前記プログラムカウンタに接続されたキャッシュ制御ロ

10

20

30

40

50

ジックとを更に含み、
 前記キャッシュ制御ロジックは、
 前記プログラムカウンタに記憶されたアドレスに対応する命令が前記対応する命令メモリに記憶されているかどうかを決定し、
 前記SIMD/MIMDレジスタが複数命令複数データ (MIMD) モードを示す場合に、もし、前記プログラムカウンタに記憶されているアドレスに対応する命令が前記対応する命令メモリに記憶されているならば、前記スイッチマトリックスを介して前記対応する命令メモリにアクセスして、命令を呼び出し、
 前記SIMD/MIMDレジスタが複数命令複数データ (MIMD) モードを示す場合に、もし、前記プログラムカウンタに記憶されているアドレスに対応する命令が前記対応する命令メモリに記憶されていないならば、前記プログラムカウンタに記憶されているアドレスに対応する命令を前記外部メモリに前記転送コントローラを介して要求し、
 前記SIMD/MIMDレジスタが単一命令複数データ (SIMD) モードを示す場合には、前記第1のプロセッサ以外のプロセッサの各々の前記キャッシュ制御ロジックはディスエーブルにされる請求項4に記載のマルチプロセッサシステム。
 【請求項6】前記複数のm個のメモリは、前記n個のプロセッサの各々に対応するメモリの所定のセットから成り、プロセッサの各々に対応するメモリの前記セットは、前記統一メモリ空間内で対応する基底アドレスを有し、
 前記n個のプロセッサの各々は、前記プロセッサの固有の識別子を記憶している複数の読み出し専用ビットを備える識別レジスタを有し、前記固有の識別子は、メモリの前記対応するセットの前記対応する基底アドレスに関連する請求項1に記載のマルチプロセッサシステム。

【発明の詳細な説明】

【産業上の利用分野】

この発明はマルチプロセッサシステム (多重処理システム) に関し、特に複数個のプロセッサと多数の異なるメモリアドレス空間とを相互接続することが可能なマルチプロセッサシステム及びその使用方法に関する。

【従来の技術】

コンピュータやプロセッサの世界では演算能力をより大きくし、演算速度を高めることが間断なく求められており、このような状況においては複数個のプロセッサを組み合わせることで互いに並列動作させることができるシステムが必要となる。

画像を生じさせてデータに対する各種の操作を行い、画像表示されたデータ及び記憶データの表示を制御する画像処理システムはどうしても大量の演算と記憶を必要とするが、このような画像処理システムは、異なるプロセッサが異なるタスクを同時並行的に実行する多重処理の適用対象の第1の候補である。この場合、プロセッサ

は、全部が同じ命令により動作するがデータは各種のソースから取り込む単一命令多重データ (SIMD) モードで同時動作させることもできるし、各プロセッサが異なる命令セットにより異なるソースからのデータを処理する多重命令多重データ (MIMD) モードで同時動作させることもできる。しかし、異なる動作モードについてはそれぞれ異なる構成が必要となる。

【発明が解決しようとする課題】

そこで、この技術分野においては、マルチメモリを備えたマルチプロセッサを取り扱うシステムであって、それらのプロセッサがいくつかの異なる命令セットを処理している時であっても1つまたは2つ以上のプロセッサがマルチメモリのすべてのメモリより得られるアドレス空間を同時に使用することができるようなシステムが求められている。

さらに、SIMDモードでもMIMDモードでも相互に切り換えて使用可能とするスイッチを具備した構成の多重処理システムないしはマルチプロセッサシステムが求められている。

後述のこの発明の一実施例のところに記載されている画像処理装置のような複雑なシステムにおける大きな相互接続の問題を解決する1つの方法は、プロセッサ全体をそっくり単一デバイスとして構成することである。これは概念的には簡単に達成することができるように見えるかもしれないが、実際には問題はかなり込み入っている。

まず第一に、情報の効率的な移動ということを充分考慮に入れると共にシリコンチップの貴重なスペースを節約したアーキテクチャが創出されなければならない。そのアーキテクチャは、いったん作り上げると、異なるアプリケーションが生じる毎それに合わせて変更、改変することは容易にはできないので、非常に高度のフレキシビリティが確保されるようにしなければならない。次に、このようなシステムの処理能力は大きなものになるから、チップに対する情報の出し入れにおいてもかなり高いバンド幅が必要となる。それはチップ1個に固着し得るリード線数が物理的に限定されるためである。

また、イメージプロセッサのようなパラレルプロセッサシステムは、広範にわたり且つ絶えず変化する使用基準を満足するに足る充分なシステムの柔軟性を確保しつつ全体をそっくり単一のシリコンチップ上に形成するような設計が望まれる。

さらに、SIMDやMIMDのような種々のモードの動作に容易に適合させ得ると共に、チップ内外間の効率的なデータのやりとりにも適合させ得るようなプロセッサメモリインターフェースを有するシングルチップ型のパラレルプロセッサを構築することが望まれる。

【課題を解決するための手段】

この発明によれば、これらの問題は、画像処理及び図形 (グラフィックス) 処理を扱う多重処理システムを設

計するに際し、すべてのプロセッサをどのメモリとでもあらゆる構成により相互接続してそれらの間でデータをやりとりさせることのできるクロスバスイッチを案出し、このシステムで n 個のバラレルプロセッサを m 個 (ただし $m > n$) のメモリに接続し得るようにしたことにより解決される。

この発明によれば、高度の柔軟性、融通性を持つシングルチップ型イメージプロセッサの製作に付随する特有の問題点が個々のプロセッサと個々のメモリとの間にマルチリンク・マルチバスクロスバスイッチを設けるというアーキテクチャにより解決され、このアーキテクチャが高密度スイッチの設計と相俟って、すべてのプロセッサのすべてのメモリに対するフルアクセスを可能とする。このクロスバスイッチは、それぞれ異なる機能に用いられる長さの異なるリンクを具備して構成され、これによってスペースの節約を図ると共に、なおかつ高度の使用上のフレキシビリティを確保し得ようになっている。

この発明の一実施例においては、転送プロセッサはオンチップ/オフチップのデータのやりとりを制御すべく動作させる一方、マスタープロセッサを共通メモリとのデータのやりとりを制御するのに使用する。動作原理としては、すべてのプロセッサが複数個の中のどのメモリにもアクセスすることができ一方、一部のメモリはもっぱら個々のプロセッサ用の命令を扱うために使用されるようになっている。

この発明のマルチプロセッサシステムは、メモリソースから供給される命令セットにより各々動作して1つまたは2つ以上の各々意のアドレス指定可能空間を有するアドレス指定可能メモリに対するデータの出入移動に依拠する多数の異なるプロセスを制御することが可能な n 個のプロセッサによって動作し、且つこのマルチプロセッサは上記メモリに接続され且つ上記プロセッサに接続されたスイッチマトリクスを有し、このスイッチマトリクスをプロセッササイクル単位ベースで選択的にかつ同時にイネーブル (動作可能) 化して上記のいずれかのプロセッサといずれかのメモリとを相互接続し、上記メモリとこれに接続された上記プロセッサとの間で1つまたは2つ以上のアドレス指定可能メモリ空間からの命令セット及びそれ以外のアドレス指定可能メモリ空間からのデータをやりとりさせるようになっている。

【実施例】

この発明の実施例のシステムのオペレーションについて説明する前に、まず第5図及び第6図に示すような従来技術における並列処理システムについて説明しておく。

第5図は、単一のメモリ55をアクセスするプロセッサ50～53を具備したシステムを示す。図示のシステムは一般にシェアドメモリ (共用メモリ) システムと呼ばれるもので、すべてのバラレルプロセッサ50～53が同じメモ

リ55に対してデータを共有的に出し入れする。

第6図は従来技術によるもう1つの並列処理システムの例を示し、このシステムではプロセッサ60～63に対してメモリ65～68が一对一の関係で分散的に配設されている。この分散メモリ型のシステムでは、複数個の各プロセッサが各々のメモリを並列にアクセスするので、動作時プロセッサ間のメモリ競合は起こらない。第5図及び第6図に示すようなシステムオペレーションの構成は、以下にも説明するように、ある特定の形態の課題を扱うのに適しており、それぞれそのような特定形態の課題に合わせて最適化されている。そして、従来は共用型あるいは分散型のいずれかになるようにシステムを作るという傾向があった。

これに対し、最近では処理要求がより複雑化しかつ動作速度の重要度が増すにつれて、共用メモリモードで実行するのが最適のオペレーション、また分散メモリモードで実行するのが最も良いものを含めて広範にわたるオペレーションを取り扱い得る能力がこの種のシステムにとって重要になってきた。この発明によれば、第1図及び第2図に示すような構成によってシステムが共用メモリモードでも分散メモリモードでも並列処理動作を行うことができるようにすることにより上記の能力を具備したシステムが達成される。また、これらのどのモードにおいても、SIMDやMIMDのような各種の処理方式が実行可能である。以下、この発明を実施例によりさらに詳細に説明する。

マルチプロセッサとメモリの相互接続

第1図に示すように、この発明のマルチプロセッサシステムはバラレルプロセッサ (並列プロセッサ) 100～103及びマスタープロセッサ12を有し、これらのプロセッサはクロスバスイッチと呼ばれるサイクルレートローカル接続網のスイッチマトリクス20を介して一連のメモリ10に接続されている。以下に明らかにするように、このクロスバスイッチは、各特定のオペレーションのために必要となる都度、分散メモリ構成と共用メモリ構成とを色々に組み合わせることができるようサイクル単位ベースで動作してこれら複数個のプロセッサと複数個のメモリを様々に組み合わせる。また、一部のプロセッサ群が一部のメモリについて分散モードで動作している時、これと同時に他のプロセッサがある特定のメモリを対象に互いに共用モードで動作するということが可能であるが、これについても以下に説明する。

第2図に示すこの発明のマルチプロセッサシステムでは、4つのバラレルプロセッサ100, 101, 102, 103がスイッチマトリクス20を介してメモリ10に接続されており、スイッチマトリクス20はこの場合分散バスとして描かれている。また、転送プロセッサ (TP) 11及びマスタープロセッサ (MP) 12もクロスバスイッチ20を介してメモリ10に接続されている。マスタープロセッサ12はバス171及び172を介してそれぞれデータキャッシュメモリ13及

び命令キャッシュメモリ14に接続されている。パラレルプロセッサ100乃至103は、以下に説明するように、これらのプロセッサ同士並びにマスタープロセッサ12及び転送プロセッサ11との間で通信する（信号をやりとりする）ことができるよう通信バス40を介して相互に接続されている。転送プロセッサ11はバス21を介して外部メモリ15と通信することができる。

また、第2図において、フレームコントローラ170はバス110を介して転送プロセッサ11と通信する。フレームコントローラ170は、以下に説明するように、画像入出力部または手段を制御するのに用いられる。画像入力部としては、例えば、ビデオカメラがあり、出力部は例えばデータディスプレイである。画像入出力は、その他如何なる手段でも使用可能であるが、その方法については追ってより詳しく説明する。

このようにクロスバスイッチ20を分散型にすると通信ボトルネックを軽減するのに役立ち、システム各部間における通信の流れを容易にすることができる。このクロスバスイッチはプロセッサ及びメモリと共に単一チップ上に集積回路化されており、これによってもシステムの各素子間の信号伝送、通信をさらに改善することができる。

また、チップ上の回路形成、集積化は数層にわたって行われるので、スイッチマトリクスの素子も相異なるいくつかの層に分かれる場合があるということも重要である。クロスバスイッチを図示する場合、縦横の線によってマトリクス状に描かれるが、実際はこれらの線は互いに空間的に隔てられて全部同一方向に配設されているに過ぎないこともある。従って、スイッチマトリクスのリンクについて言う場合、縦線及び横線という用語は互換的に使用可能であり、同じ平面あるいは異なる平面内で互いに空間的に隔てられた線を指すものとする。

ところで、この発明のマルチプロセッサシステムは多種類のモードで動作することができ、その1つは単一の命令ストリームが2つ以上のパラレルプロセッサに供給され、各プロセッサは同じメモリまたは異なるメモリにアクセスしてデータに対する処理を実行することができる単一命令多重データ（SIMD）モードである。もう1つは多重命令多重データ（MIMD）モードであり、このモードでは複数のプロセッサが通常異なるメモリから供給される複数の命令に従って動作し、同じメモリまたは異なるメモリデータバンクからのデータを処理する。この発明のマルチプロセッサシステムが動作可能なモードはこれらの2つの動作モード以外にも沢山あり、追って明らかとなるように、このシステムは必要に応じて周期的に動作モードを切り換えて異なる命令ストリームの異なるアルゴリズムを処理することも容易である。

第1図に戻って、マスタープロセッサ12はクロスバスイッチ20を介してメモリ10に接続されている。やはりクロスバスイッチ20に接続された転送プロセッサ11はバス

21を介して外部メモリ15に接続されている。また、メモリ10にはいくつかの独立メモリ及び後述する如くプロセッサ相互接続バス（通信バス）40との関連において用いられるパラメータメモリが設けられている。第2図にはパラメータメモリは1つしか示されていないが、実際にはこれは各プロセッサ毎に1つずつ複数のRAM（ランダムアクセスメモリ）で構成してもよく、そうすることによって通信、信号伝送の効率をより高めることができる共に、複数のプロセッサがこれらのRAMに対して同時に通信することが可能となる。

第4図は第1図及び第2図をより詳細に示すもので、図示のように、4つのパラレルプロセッサ100～103は通信バス40により相互に接続されると共にクロスバスイッチマトリクス20によってメモリ10にも接続されている。このクロスバスイッチの各叉点は左下角の0-0を起点とする座標により称呼される。この場合、最初の数字は縦列の番号（横座標）を表す。従って、左下角の叉点が0-0であるから、同じ最下行のそのすぐ右隣の叉点は1-0となる。例えば叉点1-5等、各叉点は後出の第19図に詳細に示すような構成を有する。第4図において、パラレルプロセッサ103等の各パラレルプロセッサはグローバルデータ接続部（G）、ローカルデータ接続部（L）及び命令接続部（I）を有する。これらの接続部はそれぞれ異なる目的に用いられるが、これについては以下に詳細に説明する。簡単には、例えばグローバル接続部はプロセッサ103をメモリ10の中のどのメモリにでも接続可能とするためのもので、それらのメモリからデータを取り込むのに用いることができる。

これらの各パラレルプロセッサのローカルメモリポートは、それぞれ、各プロセッサの対向位置にある縦方向または縦列スイッチマトリクスリンクに対応するメモリのみをアドレス指定することができる。即ち、パラレルプロセッサ103の場合は、クロスバスイッチマトリクス20の縦リンク0、1及び2を使ってメモリ10-16、10-15及び10-14をアクセスし、MIMDモードにおけるデータ転送を行うことができる。さらに、MIMDモードの時、メモリ10-13はプロセッサ103に命令ストリームを供給する。追って説明するようにSIMDモードにおいてはプロセッサ用の命令はすべてメモリ10-1より供給される。そのため、命令メモリ10-13はデータ用として使用可能である。その場合、クロスバスイッチマトリクス20は縦リンク4を介してローカルメモリポートよりアクセスすることができるよう構成し直すが、そのやり方については後述する。

第4図に示すように、パラレルプロセッサ100～103は各々色々なメモリにアクセスすることができるよう特定のグローバルバスと特定のローカルバスを有している。即ち、パラレルプロセッサ100はクロスバスイッチマトリクス20の横リンク2よりなるグローバルバスを有し、パラレルプロセッサ101はスイッチマトリクス20の横リ

リンク 3 よりなるグローバルバスを有する。また、パラレルプロセッサ 102 及び 103 はそれぞれスイッチマトリクス 20 の横リンク 4 及び 5 よりなるグローバルバスを有する。

各パラレルプロセッサに接続されたローカルバスは全部でスイッチマトリクス 20 の横リンク 6 を共用する。ただし、この横リンク 6 は、図示の如く、3 つのスリーステートバッファ 404、405 及び 406 により 4 つの部分に分けられている。これによって横リンク 6 は各プロセッサの各ローカル入力部が異なるメモリをアクセスすることができるよう効果的に分離される。この構成はシリコンチップ上のレイアウト面積をより効率的に利用するために案出されたものである。上記バッファはこれにより分割される複数の部分を後述の如く必要に応じて 1 つに接続し、プロセッサ間でのデータの共通伝送を可能にする。この構成によれば、メモリ 10-0、10-2、10-3 及び 10-4 からのデータをパラレルプロセッサ 100~103 のどれにでも伝送することができる。

マルチプロセッサシステムが MIMD モードで動作している時は、パラレルプロセッサの命令ポート、例えばプロセッサ 103 の命令ポートは又点 4-7 を介して命令メモリ 10-13 に接続される。このモードでは、又点 4-2、4-3、4-4、4-5 及び 4-6 は 4-1 共々ディスエーブル（動作不能）化される。また MIMD モードにおいて又点 4-0 は動的作動性の又点であり、これによって転送プロセッサも必要に応じて命令メモリ 10-13 にアクセスすることができる。これと同じ手法の動作が又点 9-7（プロセッサ 102）及び又点 14-7（プロセッサ 101）についても可能である。

このマルチプロセッサシステムが SIMD モードで動作中は、又点 4-7 はイナクティブ（非動作）状態であるが、又点 4-2~4-6 はアクティブ（動作状態）化可能であり、これによってメモリ 10-13 のデータはクロスバススイッチマトリクス 20 の縦リンク 4 を介して全部のパラレルプロセッサ 100~103 が利用可能になる。同時に、SIMD モードでは、バッファ 401、402、及び 403 がアクティブ化され、これによって命令メモリ 10-1 は、プロセッサ 100~103 の全部が各々の命令入力ポートを介してアクセスすることができる。例えば、バッファ 401 がアクティブ化され、バッファ 402 及び 403 はイナクティブであるとする、プロセッサ 100 及び 101 は命令メモリ 10-1 を共用して SIMD モードで動作することができる一方、プロセッサ 102 及び 103 はこれとは独立にメモリ 10-13 及び 10-9 の命令を MIMD モードで実行する。

又点 18-0、13-0、8-0 及び 3-0 は転送プロセッサ 11 をいずれかのパラレルプロセッサの命令入力ポートに接続可能とするためのもので、この接続は例えばキャッシュミスが生じたような場合に転送プロセッサがこれらのパラレルプロセッサにアクセスすることができるようにする等、いろいろな目的に利用可能である。

第 7 図はメモリ 55 及び 55A に関しては共用モード（前出、第 5 図）で動作し、メモリ 65~68 に関しては分散モード（前出、第 6 図）で動作するパラレルプロセッサ 100~103 のオペレーションの構成概念を示す。図示のようなパラレルプロセッサのフレキシブルな接続構成を達成する具体的方法を以下に説明するが、これは以下の説明からも明かなように、複数のリンクを具備したクロスバススイッチマトリクス 20 の動作に基づくものであって、このスイッチマトリクスはそれらのリンクが又点で個々に作動して所望の様々な接続構成をもたらすようになっている。

クロスバススイッチの動作説明に入る前に第 3 図に示すようなもう 1 つのシステム構成について考察しておく方が良いと考えられる。図示の構成にあっては、バス 34 を各々第 1 図及び第 2 図で説明したような構成を有する一連のプロセッサ 30~32 と接続された状態に設定することができる。図示の外部メモリ 35 は、第 1 図、第 2 図のメモリ 15 と同様単一メモリよりなる。もちろん、この外部メモリはシステムに対しローカル配設されたもの及びリモート配設されたものを含め、一連の複数の個別メモリで構成してもよい。第 3 図の構成は、何種類ものプロセッサをそれらのすべてのプロセッサが統合されたアドレス指定能力を有する単一のグローバルメモリ空間にアクセスするものとして本願におけるイメージシステムプロセッサ [(ISP)] と共に 1 つに統合するのに用いることができる。また、この構成は、すべてのパラレルプロセッサが各々独自のオペレーションを実行している間に信号をやりとりし合うことができ且つ秩序を保つことができるようにバス 34 を介してメモリアクセスする統合コンテンション型構成とすることもできる。ホストプロセッサ 33 は複数のプロセッサ 30~32 の間の競合整理という問題の一部を受持つて、バス 34 上のデータフローの秩序を確保するよう機能する。

画像処理

画像処理では、ある画像に対して実行可能ないくつかのレベルのオペレーションがある。これらのレベルは、単にデータをその内容を理解することなく伝達して基本オペレーションを実行するだけのレベルを最下位レベルとするいくつかの異なるレベルよりなると考えることができる。この最下位レベルのオペレーションは例えば画像から異質な斑点を除去する動作等であり、これより少し上のレベルは例えばデータのある部分が円を表すということを認識するものの、その円が人間の顔の一部分であるということを十分に理解することはできないというようなレベルである。画像処理におけるもう 1 つ上のレベルのオペレーションとしては、例えば、色々な円やその他の形が人間の顔のイメージあるいはその他のイメージを形成するということを理解して画像を処理すると共に、そのような情報を様々な形で利用する等のオペレーションがある。

画像処理におけるこれらの各レベルのオペレーションはプロセッサをある特定のモードで動作させることにより最も効率的に実行される。従って、画像全体を理解しようとせずにローカル的にまとめられたデータに対してオペレーションを実行する時は、通常、全部または一群のプロセッサが単一命令と多重データソースからのデータにより動作するSIMDモードを用いる方がより効率的である。これに対して、全体画像を理解するためにそのいろいろな特徴からピクセル（画素）データが要求されるようなより高いレベルのオペレーションは、各プロセッサがそれぞれ個別の命令によって動作するMIMDモードで実行するのが最も効率的である。

ここで重要なのは、システムがSIMDモードで動作している時、全ピクセルイメージを単一命令ストリームにより色々なプロセッサを通して処理することができるということである。そのオペレーションは、例えば、画像全体をクリーニングする処理の場合や、様々なコーナー（角部）や縁部を浮き立たせるために画像強調する処理の場合等である。そして、すべての画像データはSIMDモードでプロセッサを通過するが、各時点において画像の種々異なる部分からのデータを異なる目的に合わせて異なるやり方で同時に処理することは不可能である。SIMDオペレーションの一般的特徴は、どの期間においても処理中のデータ量は画像全体に対して比較的少量であるということである。逐次後の段階ではより多くのデータが同じやり方で処理される。

これと異なり、MIMDモードでは画像の色々な部分からのデータが一部は異なるアルゴリズムを用いて同時に処理される。このモードのオペレーション構成では、所望の結果を得るのに異なるデータに対して異なる命令が同時に実行される。その簡単な例としては、例えば、多くの異なるSIMDアルゴリズム（クリーン（クリーニング）、エンハンス（強調）、エキストラクト（抽出））等のようなアルゴリズムが同時に処理される場合、あるいは多くの異なるプロセッサでパイプライン処理される場合等が考えられる。MIMDのもう1つの例としては、一意の演算機能あるは論理機能を用い一つも同じデータフローについていくつかのアルゴリズムを実行する場合等を挙げることができる。

第8図及び第9図はそれぞれ従来技術によるSIMD及びMIMD構成のプロセッサを各々のメモリと共に示している。図示の構成は画像処理に好都合なSIMD/MIMDのトポロジー（接続形態）である。このシステムのオペレーションモードについては追って第59図乃至64図を参照しつつより詳しく説明する。概して言う、第8図のデータバス80は第60図のデータバス6010、6020、6030及び6040に相当し、第9図のプロセッサ90は第59図のプロセッサ5901、5911、5921、5931に相当する。これらのデータバス用のコントローラ（第60図の6002）は第8図では省略されている。

SIMD/MIMD構成変更可能型システム

第10図はこの発明のSIMD/MIMD構成変更可能型（構成可変型）システムのトポロジーを示し、図示のシステムでは、すべてサイクル単位ベースで、複数のパラレルプロセッサをクロスバスイッチマトリクス（ローカル及びグローバル相互接続網）20を介して一連のメモリ10に接続することができると共に、転送プロセッサ11を介して外部メモリ15に接続することができる。

この場合、MIMDトポロジーでオペレーションを実行する際の問題の1つは、実際のデータフローがシリアルベースで行われるかトポロジー的にエミュレートされるSIMDモードに比べて高いバンド幅を必要とする場合があるということである。この点、SIMDモードでは、データは一般に1つのプロセッサから次のプロセッサへといくつものプロセッサを逐次通って流れる。このことは有利な面もあるが、問題でもある。これが問題となるのは、処理のある点に到達するためには画像のデータをすべて処理しなければならないという点においてであるが、SIMDモードではこの処理はシリアル的に行われる。しかしながら、MIMDモードでは、共用メモリはデータが到着する毎にシリアルベースでしかアクセスできないSIMDのオペレーションに対して、個々のメモリのデータはサイクル中いつでも取り込むことができるので、上記の問題は解消される。

しかしながら、MIMDモードは、プロセッサ間通信（IPC）を行うことを要求される場合にはオペレーション上のボトルネックがある。と言うのは、この場合、1つのプロセッサがデータをメモリに書き込んだ後に他のプロセッサがそのことを認識してそのメモリにアクセスするようにしなければならないからである。この動作にはかなりのサイクルタイムが必要なこともあるため、大量のピクセルデータを伴う大きな画像は相当長い処理時間が要求され得る。これは大きな難点である。上にも述べたように、第10図の構成によれば、これらの問題は解消される。それは、クロスバスイッチは必要に応じてサイクル単位ベースで複数のプロセッサをまとめて相互接続して、ある期間単一命令により動作させるか、または第1のメモリに記憶されているデータが、他のプロセッサが1サイクルあるいは一定期間これと同じメモリに接続されている間、その第1のメモリに保持されるよう、これらのプロセッサをそれぞれ独立に動作させるように相互接続することができるからである。要するに、従来技術では、データを多数のプロセッサにアクセスさせるにはそのデータをメモリからメモリへ移動させなければならないことがあったが、この発明のシステムでは、データは一貫してメモリ内に保持しつつメモリに対する接続構成は必要に応じて切り換えることができる。これによれば、プロセッサとメモリのオペレーションにおける完全なフレキシビリティを確保し得ると同時に、データ転送リソースの最適活用が可能である。

ここで、種々のSIMD及びMIMDモードにおけるデータ処理の具体例を第12図及び第13図を参照しつつ説明する。第12図には一連のピクセル（画素）0～nを有する画像125が示されている。図示の画像は1行にピクセルが4つしかないように描かれているが、これは例示説明のためであって、通常は画像は例えば1000行からなり、各行には1000個ものピクセルがある。また、画像の行数及び各行中のピクセル数は各時点毎に変化し得るが、ここでは、1行は4ピクセルよりなるものと仮定して説明する。メモリ124内におけるこれらのピクセルを表す1つの方法は、上から下に向けてピクセル0、ピクセル1、・・・、ピクセルnと記されたメモリ124の各アドレス指定可能空間にそれぞれこれらのピクセルが入っているものとみなすやり方である。以下に述べるように、このメモリは単一のメモリでも一連の複数のメモリで構成されたものであってもよいということはもちろんである。複数のメモリを用いる場合は、データを各行毎に異なるメモリに記憶するような構成とすることも可能である。

ここで、全部のピクセルまたはピクセルの何らかのサブグループについてのデータに関し、そのデータをすべて同じ命令により処理してからメモリに戻すようなオペレーションを行なおうという場合について考える。この場合、例えばメモリ124のピクセル0の空間からのデータはプロセッサ120にロードされ、プロセッサ120から121へ転送され、そこからプロセッサ122へ、さらにプロセッサ123へ転送される。そして、この各転送動作毎に新しいデータがエントリされるというような動作が行われる。この手法を用いると、プロセッサ120～123はそれぞれデータに対して何らかの処理を行う機会を与えられ、それらのデータに対して以前に行われた処理を観察する機会を与えられる。このような動作が一巡すると、データはメモリに戻される。このサイクル動作は、画像の部分集合の全ピクセルまたは画像中の全ピクセルをシステムを通して逐次処理するよう反復実行することができる。この種のオペレーションはSIMDモードで実行するのが最も良い。

第13図には、これとは対照的なMIMDモードのデータフローが図示されている。図示のようなシステムでは、ピクセル0～3とピクセル250～500をある特定の手法で処理する一方、他の画像領域（例えば領域3で示す画像の一定領域）のピクセルは別のやり方で処理することが望ましいと考えられる。そこで、例えばプロセッサ120がピクセル0～3及びピクセル250～500を処理し、他方プロセッサ121はピクセル50～75及びピクセル2000～3000を処理するような構成とすることができる。このようにして、異なるアルゴリズムを用い、あるいはデータ内容によりプログラムフローを変えつつ同じアルゴリズムを用いて画像の各領域を処理することができる。これらのピクセルはすべて並列に処理され、種々の記憶場所に記

憶される。このようなオペレーションは、データをプロセッサ121からプロセッサ120へ移動させなければならないような場合以外は、MIMDモードの方がSIMDモードよりも高速に行うことができる。データを移動させる場合は、メモリバンク内でデータ移動を行わなければならないと考えられる。このようなプロセッサ間データ移動は、例えば、ある領域からの処理されたデータが他の領域からのデータをどのようにして処理するかを決定する上において重要である場合、あるいは画像全体が表すものを正確に決定するために重要である場合等に必要となり得る。鼻をつかんで象の形を決定することは困難であるが、これと全く同様に、他の色々な部分のピクセルデータにアクセスすることなくして画像から有意な情報を得ることは非常に困難である。

次に、第14図には、この発明を用いたシステムの構成が描かれている。図示のシステムにあっては、クロスバスイッチマトリクス20はサイクル単位ベースで、プロセッサ100～103がメモリ10の個々のメモリ素子M1～M4をアクセスできるようにする。第14図の構成では、SIMDモードに関連して第12図により説明したオペレーションが可能で、メモリ素子M1～M4のデータは動かさずに、プロセッサとの接続構成を切り換えるようなオペレーションが実行される。このプロセスは、プロセッサがある瞬間に実際に使用するより多くのメモリ素子を用いることにより流れの連続性が改善されている。このように、データはこれらの「エキストラ」メモリ素子から入出力できるので、これらエキストラメモリ素子はシステムのオペレーションの流れの中に組み入れてサイクル動作させることができる。このような構成にあって、データ入力用メモリ素子及びデータ出力用メモリ素子は、サイクル単位ベースで異なるメモリ素子が使用され得る。データ入力用メモリ素子及びデータ出力用メモリ素子はクロスバスイッチを介して切り換えられるので、どの位置のメモリ素子であってもよい。このように、このシステムでは、データをメモリ間で移動させる代わりにプロセッサの接続構成が逐次切り換えられる。

次に、第15図には、プロセッサ100～103をクロスバスイッチ20を介して複数のメモリに接続するようにしたMIMDモードの構成が示されている。このシステムでは、通常、図示のような接続関係が数サイクル持続するため、各プロセッサは各メモリにある期間に亘って接続されることになる。その必要がないのであれば、それは最も典型的なMIMDモードのオペレーションの場合であろう。いずれかのプロセッサまたはプロセッサ群が第15図のMIMDモードで動作する場合、クロスバスイッチ20は、あるメモリ素子からのデータを直ちに他のどのプロセッサでも使用できるようにして、そのデータをこれら他のプロセッサに循環させるか、あるいは1回ベースで処理できるようにサイクル単位ベースで動作可能である。

接続構成可変式プロセッサ間通信

第16図は、複数のプロセッサが互いにやりとりしなければならない時にシステムがMIMDモードで動作している時のプロセッサ間通信の接続構成を示す。パラレルプロセッサ100のようなプロセッサはクロスバスイッチ20を介して共用パラメータメモリにメッセージを送る一方、これと同時にパラメータメッセージが待機状態にあると言うメッセージ（割込み）を行先プロセッサに伝える。ここでは、行先プロセッサはプロセッサ102等上記プロセッサ100以外のどのプロセッサでもよく、クロスバスイッチ20を介して共用パラメータメモリにアクセスし、メッセージを取り除く。すると、行先プロセッサは例えば受け取ったメッセージに従って自ら再構成あるいは構成変更する等の動作を行うことができる。この構成変更は特定のシステムオペレーションモードを得るべく内部的に行うこともできれば、ある期間中はどのメモリにアクセスすべきでどのメモリにはアクセスすべきではないということに関する命令の形であってもよい。

メモリアクセスの問題（コンテンション）は、プロセッサは他のプロセッサがあるメモリを長期間使用している時そのメモリにアクセスしようとして多大の時間を浪費してしまうことがあるので、非常に重要である。システムの効率的なオペレーションは通信リンクを介してのプロセッサ間結合なしでは達成が極めて困難である。

プロセッサ間でやりとりされるもう1つの形態のメッセージはそれらのプロセッサの同期に関するものである。これらのメッセージ及び同期を確保するための詳細なやり方については以下に説明する。前出の第2図は、複数のプロセッサがサイクル単位ベースで同期、メモリ及びクロスバスイッチのアロケーションを制御すべくそれらのプロセッサ間で割込みまたはポーリングを行うために相互接続されるシステムの全体構成をしめす。

メモリのより効率的な活用に役立つのはクロスバスイッチの外部で機能する通信リンクである。オペレーションモード間、例えばSIMDとMIMDモードの間の切り換えを行うのに必要なサイクル数はそれ以外に実行しなければならない他のオペレーションの量によって決まる。これらの他のオペレーションとは、例えば、いくつかの命令メモリにコードをロードしたり、データを後続のオペレーションのためにデータメモリにロードする等のオペレーションである。外部通信リンクは、ある特定のプロセッサがどのメモリにアクセスする可能性があるかを確定すると共に、すべてのプロセッサにそれらのプロセッサのメモリアクセスの可否を知らせて、アクセスが拒絶されているのにそれらのプロセッサが並んでアクセス待機するようなことが起こらないようにすることによって上記のメモリの効率的活用を支援することができる。

プロセッサ間の命令のやりとりは割込みによりまたポーリングによって行うことができる。割込みは、データを共用メモリ内の特定のメッセージ記憶場所に仕向ける

フラグを付して送出する周知の割込み構成のいずれかによって行うこともできれば、プロセッサ内部でポインタベースで直接処理するようにしてもよい。どのプロセッサがどのメモリにアクセスできるかをサイクル単位ベースで確定する能力は、システムが、データを特定メモリ内に終始保持しつつ、そのデータにアクセスできるプロセッサを次々に切り換えるやり方によりMIMDモードで動作し得る能力を確保する上において重要である。この構成を用いるならば、メモリがプロセッサに対して固定された関係にある場合にデータをメモリからメモリへ移動させるのに必要な時間サイクル数を顕著に減らすことができる。上記の通信リンクにはマスタープロセッサが含まれている。

転送プロセッサ

第1図及び第2図並びに第57図に示す転送プロセッサ11は外部メモリと複数の内部メモリの間でデータを転送する。転送プロセッサ11は、パラレルプロセッサのいずれかまたはマスタープロセッサが転送プロセッサ11に特定のピクセルまたはピクセル群あるいはデータ群に関するデータを供給するよう要求することができ、且つこれに対して転送プロセッサ11が必要なデータをそれ以上のプロセッサ介入命令なしに外部メモリと内部メモリの間で転送することができるように、パケット要求に応じて動作するよう設計されている。そのため、転送プロセッサ11は自律的に動作することができ、且つどのプロセッサによる監視をも受けることなくシステムの内外にデータを転送することができる。転送プロセッサ11はクロスバスイッチマトリクス20を介してすべてのメモリに接続されて、これらのメモリへのアクセスにおいて様々なリンクと競合するようになっている。いずれか特定のリンクについては、転送プロセッサ11に最も低い優先順位を割当て、メモリへのアクセスを他のプロセッサがそのメモリにアクセスしていない時に行うようにしてもよい。転送プロセッサ11により転送されるデータはピクセルを処理するためのデータのみならず、システムを制御するための命令ストリームも転送される。これらの命令ストリームはクロスバスイッチ20を介して命令メモリにロードされる。この転送プロセッサ11は、データ転送という目的を達成するためにハードウェアとソフトウェアを組合せた構成を用いることもできる。

マスタープロセッサ

マスタープロセッサ（第29図参照）は、転送プロセッサの制御や色々なプロセッサの間のインタラクションを含め、システム全体のスケジューリング及び制御を行うために用いられる。マスタープロセッサはクロスバスイッチを介してすべてのメモリに接続可能であり、またチャンネル上の他のプロセッサとも相互接続される。マスタープロセッサはピクセル情報およびその情報を取得する目的によってデータの型を制御すると共に転送プロセッサがデータを取得する手法を制御する。従って、画像

の各領域は、走査の目的によって異なるモードにより走査することができる。この動作はパラレルプロセッサと連携して動作するマスタープロセッサにより制御される。パラレルプロセッサは、やはりオペレーションの目的により、単独であるいはマスタープロセッサと共同で各々転送プロセッサを制御するようにしてもよい。

クロスバスイッチに対するメモリのコンテンションは、パラレルプロセッサには比較的高い優先権を与え、マスタープロセッサには比較的低い優先権を与えると共に、転送プロセッサの特定リンク上の特定メモリに対する優先権は3番目かあるいは最下位となるよう構成することができる。

画像処理システムで実行されると思われる典型的なオペレーションあるいはアルゴリズムの例が第11図にリストしてある。中でもより典型的なオペレーションは光学的文字認識、目標認識あるいは運動認識であろう。これらの各オペレーションにおいては、各々に付随する画像処理動作は、実行されるオペレーションの型により制御されることになろう。

第11図において、一般にパラレルプロセッサにより実行される種類のオペレーションは破線1100の下側に示されており、一般にマスタープロセッサにより実行される種類のオペレーションは破線1100の上側に示されている。このようなオペレーションの構成はマスタープロセッサとパラレルプロセッサの間で任意に分割されるが、図示のような各種のオペレーションを遂行するのに必要なオペレーションの種類によってマスタープロセッサとパラレルプロセッサのどちらがより適切であるかを仕分けすることができる。

画像から出発してオペレーション階層のより高い所で処理が行われる画像処理の一例を第11図により説明すると、画像はまずイメージエンハンスメント（画像強調）のボックス1111に受け入れられる。状況によっては、ボックス1112及び1113による画像の圧縮または圧縮解除が必要になる。次に、画像はエッジ抽出1109、ラインリンケージ（線結合）1107、角または頂点の認識1105、ヒストグラム1110、統計学的性質1108及びセグメンテーション1106等のボックスを通じて図の上方に移動させられ、必要に応じてこれらのボックスにより処理される。これらのボックスはすべてスキップ可能であり、画像を直接テンプレートマッチング1102のボックスに与えて、識別1101のボックスにより画像を識別させるようにすることもできる。このような識別を行う方法は種々あるが、画像毎にそれらの方法を全部行う必要はなく、またそれらの方法はすべて個別のアルゴリズムまたは方法として周知である。

エンハンスメントボックス1111は基本的には画像をきれいにし、異質な信号を取り除き、線のような画像の細部を強調するプロセスである。ボックス1109のエッジ抽出は画像の中のエッジの原因や有無を判定するプロセス

である。ボックス1107では、画像から抽出されたすべての線が結合され、より長い線が形成される。このボックスでは、その後データ中の不一致により生じた異質な末梢の線が除去される。角／頂点のボックス1105は画像の角部や頂点がある場合にその位置を決定するアルゴリズムである。いったんこれらの幾何学的形状の特徴が見出されたならば、次にボックス1104のグルーピング及びラベリングというプロセスを用いて円、矩形というような物体の大まかな分類を行うことができる。

10 この段階では、ボックス1111では一般に画像全体が処理されるのに対して、画像の比較的小さな領域に的を絞ってオペレーションが行われる。イメージエンハンスメント1111を経た後の上記と異なるもう1つの進行経路は、ピクセルの輝度についてヒストグラム（ボックス1110）のような統計学的分析を行うものである。ヒストグラムの1つの目的は、論理値「1」の数或いは特定の軸線または投影線中の「1」の数を検出し、何らかの物体の存在または物体の配位について定量化するための有用な統計学的情報を得ることにある。以下、これについて

20 説明する。
ボックス1108（統計学的性質）では上記のヒストグラムより適切な統計学的性質が抽出される。その上方のセグメンテーションのボックス1106は統計学的情報を用いて様々の物体をセグメント化する。例えば、いくつかに切り離された物体は極めて容易にセグメント化することができる。次に、グルーピング及びラベリングのボックス1104に進むと、画像を構成する異なる物体の像に各々特定のラベルが付され、識別が行われる。この処理で典型的なアルゴリズムは結合子コンポーネントのアルゴリズムである。ここでもある種の幾何学的特徴、特に物体の周囲を分析することができる。また、他の形状記述子、オイラー数、及び表面の特徴に関する記述が得られ、その後のマッチングオペレーションで用いられる。ボックス1102のマッチングオペレーションのレベルでは、テンプレートまたはライブラリとして記憶された類似の情報がアクセスされ、下位レベルより抽出されたデータと対比される。これは幾何学的特徴、表面の記述でも、光学的フローの情報でもよい。この対比によりいったん一致の結果になると、それらの一致情報に統計学的に重みが付けられて、ボックス1101に示すように、ある物体が識別されたという確度が与えられる。このように物体の識別が行われると、立体映像や立体映画のような用途分野において、この世の三次元表示が可能となり、表示されたものが何の物体で、表示された世界のどこに置かれているかをはっきり示すことができるようになる。次に、このような表示画面は第11図の右側に図示されているようなグラフィックス（図形）のパイプライン処理を用いて再処理することができる。

50 まず最初のボックス1114の幾何学モデルでは、その画面の表現情報が識別（同定）される。その情報は、基本

的には位置を示す3つの座標及び形状、密度及び反射特性等のような物体の幾何学的記述である。ここで、物体の種類によっては画面の処理にいくつかの異なるルートが用いられる。例えば、簡単なキャラクタの場合は2次元変換の手法が用いられる。より複雑なものについては3次元の世界が創出されることになる。例えば、ジェスチャ入力デバイスとして用いられるコンピュータの前での手振りはこの方法を用い、ボックス1116の3次元変換の機能が実行されることになる。この機能は、3次元変換ボックス1116により3次元座標を平行移動、スケーリング（拡大縮小）あるいは回転させることにより入力を新しい座標系に変換するものである。物体は、他の物体によって閉塞される場合もある。例えば、再度手を例にとると、指の一部が他の指によって閉塞されてしまう場合があるが、このシステムの実行ではジブリティのボックス1117を用いて見えないはずの部分は無視するようになっている。その下方の陰影付立体のボックス1118では、グレースケールまたはピクセル情報を発生してスムーズな陰影を有する立体画像を得るためのプロセスが実行される。これにより得られる画像は、クリッピング処理（ボックス1120）に到るまでに他のルートを介して処理した場合に比べてよりリアルで本物そっくりに映る。ボックス1120のクリッピングでは、本来、生成中の画面の視野外に来る事物をクリップして除去する処理が行われる。

コンピュータの出力画面あるいはレーザプリンタ等に文字フォントを生じさせるような特殊な場合には、ボックス1119のフォントコンパイルの実行を用いることにより多数のサイズと形状の見事なフォントを作り出すことができる。グラフィックスプログラムにおける最終プロセスはボックス1121により実際に描画することであるが、これは単に点を描いたり、これらの点を結んだ線を描くというような簡単なオペレーションになることもある。この描画は最初のイメージエンハンスメント1111と同じレベルにあるが、この段階ではオリジナルイメージから導出されたモデルに基づきそのオリジナルイメージが合成表現あるいは合成画像として再現される。

いったんキャラクタが認識されたかあるいは運動が認識されたならば、出力を2進コードその他により得ることができ、その出力を用いて出力制御のボックス1122により同じ画像の以後の処理をこの画像処理システムと共に使用するパラレルプロセッサとマスタープロセッサのオペレーション及び組合せによって制御することができる。

一般には、第11図の破線1100の下側のボックスのオペレーションはSIMDモードで実行する方が効率的であり、膨大な量の処理を必要とする。これらは並列処理オペレーションにより実行される。破線1100の上側のオペレーションは比較的处理能力が小さくてもよく、またバンド

幅により制約される度合いも小さい。従って、これらのオペレーションは単一のプロセッサで実行される。また、これらのオペレーションに関しては、この図表で階層が上がるにつれてMIMDオペレーションによる方が望ましいであろうということも注目される。このようにSIMDオペレーションとMIMDオペレーションの守備範囲は重なり合い、両方のオペレーションモードが必要なこともしばしばある。

このマルチプロセッサシステムでは2種類のプロセッサが必要であるが、その主な理由は異なるレベルの処理が必要なためである。マスタープロセッサが実行するようなハイレベルの処理は、高精度を確保するため浮動小数点演算手法を用いることが望ましい。高精度浮動小数点プロセッサはより大きな実記憶空間を必要とし、非浮動小数点型プロセッサから動作させようとすると演算が遅くなる。従って、全部同じ種類のプロセッサ（浮動小数点型）を用いると、所与のチップ上に形成し得るプロセッサの数は少なくなり、その結果バンド幅の問題が大きくなり且つシステムの動作が遅くなるということになる。他方、ローレベル処理用のプロセッサは浮動小数点演算が不要で、高速化、小型化が可能のため、所与のチップ上により多くのプロセッサを形成することができる。このような状況にあつて、この発明のクロスバスイッチを利用したバス構成によれば、必要に応じて数種類のプロセッサを用いることが可能となり、これらのプロセッサを必要に応じてシステムに切り換え接続して各オペレーションの様々な部分を実行させることができる。

マスタープロセッサは主として情報リストやディスプレイリストのようなリストによって動作するよう設計されており、これにたいしてパラレルプロセッサはアレイによって動作するよう企図されている。ローレベルの画像処理では情報の大半は2次元アレイで記述し得るが、これよりハイレベルになると、情報は多次元座標のリストとして記述される。これらの2種類のデータ表現の取扱いは異なる処理構成が要求されるが、このこともこのシステムで互いに構成の異なるマスタープロセッサ及びパラレルプロセッサを採用したもう1つの理由である。

この実施例のマスタープロセッサはどちらかというと主に汎用計算用と考えられているRISC（限定命令セット計算機）型のプロセッサに類似の特徴を有するが、パラレルプロセッサは演算用の特殊プロセッサとして用いられる傾向にあるデジタルシグナルプロセッサ（DSP）により類似している。従って、このマルチプロセッサシステムは画像処理システムに必要なタイプの情報処理用として最適化することができる一方、なおかつデータに対し上記の両方のプロセッサを用いることにより高度の処理能力と全体的なフレキシビリティを確保することができる。

共同譲渡になる米国特許第4,577,282号及び第4,713,7

48号並びにやはり共同譲渡になる米国特許出願一連番号第025,417号(1987年3月13日出願)には、テキサスインスツルメントTMS 320型DSPプロセッサが開示されている。その詳細な背景については、テキサスインスツルメント社発行の刊行物「第2世代のTMS320ユーザーズガイド」及び「第3世代のTMS320ユーザーズガイド」に開示がなされている。上記の特許、特許出願並びに刊行物は本願中に引用がなされている。

メモリ構成

第17図は前出の第1図及び第2図により説明したような画像処理システムにおけるメモリのレイアウトの一例を示している。図示の特定のメモリサイズは特定のプロジェクトのために採用されたもので、この発明においてはメモリ及びメモリ及びメモリ容量についてどのような構成でも用いることができるということはもちろんである。メモリ10のパラメータ部はメモリ10内に組み込むこともできるし、その方が望ましい場合はスタンドアロン型メモリとしてもよい。一定条件下では、このパラメータメモリはプロセッサの通信要件次第で必ずしも設けなくともよい。

クロスバスイッチ

第18図は、クロスバスイッチマトリクス20の優先化回路群の図で、クロスバスイッチマトリクス20の各縦リンク(縦列系統)は各特定叉点内の優先化回路にラウンドロビン方式(巡回優先順位指定方式)により接続されている。各縦列系統中、横リンク(横行系統)の転送プロセッサに割り当てられた最下行に属する叉点は優先化回路群の配線には含まれていない。これは、同一縦列系統中の最下行以外のどの横行系統も選択されていない時に転送プロセッサがその縦列系統に対応するメモリにアクセスすることができるようにするためである。この優先化回路群の動作及び最下行の叉点の動作については、以下に第19図及び20図を参照しつつ詳細に説明する。

第18図は、また、パラレルプロセッサの命令入力ポートIに対応する縦列系統の特殊な情況をも示している。このパラレルプロセッサ103の命令入力ポートIに対応する縦列系統は叉点4-7を介して接続されるが、この叉点はインバータ1801を介してSIMDリード上の信号によってイネーブル化されている。これと同じ信号が同じ縦列系統の横行に対応する叉点4-1乃至4-6にもそれぞれ供給されてこれらの叉点をイネーブル化している。以下、このSIMDリード上の信号及び上記縦列系統をメモリに接続する仕方について説明する。

第19図には一例として叉点1-5の構成が詳細に示されている。この図において、側面から制御線が入る五辺形は制御スイッチで、通常FETデバイスからなる。

ここで叉点ロジックの機能性について説明する。この叉点ロジックは4つの機能ブロックを含んでおり、これらについて各個に説明する。まず第1の機能ブロックはアドレス認識ブロック1901で、バス1932を介してプロセ

ッサにより供給されるアドレスの5ビットをバス1930上にあるメモリモジュール10-15(第4図に示すように縦列1を介して叉点1-5に接続されている)の一意の5ビット値と比較する。上記バス1930上の値はアドレス空間内のメモリの記憶場所を示す。上記の比較は各ビットを個別に比較する2入力排他NORゲート1920-1924により行われる。これら5つのゲートの出力は6入力NANDゲート1910の5つの入力に供給される。このゲート1910の6番目の入力は、メモリ要求が実際に実行されており、プロセッサによるアドレス出力を実際に比較すべきであるということを示すグローバルアクセス信号1933に接続されている。この信号1933が論理値「1」で、ゲート1920-1924の出力もすべて「1」である時のみゲート1910の出力は「0」となる。この出力が「0」ということはメモリ10-15に対して有効な要求がなされているということを示す。

ここで少し本題から逸れるが、上記のアドレス認識ロジックに対して可能な変更、改変としては、例えばゲート1910に、叉点ロジック用のイネーブル信号として用いることのできる7番目の入力(イネーブルSIMD)を付加することである。このイネーブル信号が論理値「0」であるとアドレス認識ロジックがディスエーブル化され、その結果叉点全体がディスエーブル化される。この論理信号は水平バス106に接続された縦列バス4、9及び14上の叉点で用いられ、SIMDモードでこれらの叉点をイネーブル化する一方、MIMDモードではディスエーブル化する。

第2の機能ブロックはトークンラッチ1904である。この機能ブロックは、ラウンドロビン優先化の開始点を示すのに用いられる信号B1を出力する。この信号B1は、叉点1-5の下方の次の叉点ロジック(叉点1-4)の入力信号Bに接続される。(叉点1-1の信号B1は、ぐるっと回って叉点1-6の信号Bに接続され、第18図に示すような巡回式優先化構造を形作っている。)縦列バス1に対応する叉点ロジック内の唯一の信号B1だけが論理値「0」を出力し、他はすべて論理値「1」を出力する。この関係は、システム初期化時に1つの叉点のトークンラッチ1904に「0」をロードし、それ以外のトークンラッチには「1」をロードすることによってのみ達成される。このためには、1つの叉点のプリセット値信号を論理値「0」に接続すると共にその叉点についてはプリセット値信号を「1」に接続し、かつクロック5をアクティブ化する。これによって、トランジスタ1956を介してプリセット値がインバータ1946及びインバータ1945よりなるラッチにロードされる。このプリセット値は、次いで、トランジスタ1955を介してクロック2によりクロック同期され、インバータ1947及びインバータ1948よりなるラッチにロードされる。インバータ1947の出力は信号B1である。この信号は2入力NANDゲート1913の一方の入力に供給され、このゲートの他方の入力にはゲート

1910の出力が供給されている。上記ゲート1913の出力は2入力NANDゲート1914の一方の入力に供給され、このゲートの他方の入力にはゲート1911の出力が供給されている。ゲート1914の出力はトランジスタ1952を介してクロック4によりクロック同期され、前述のゲート1945及び1946よりなるラッチにロードされる。また、クロック2及びクロック4は決して同時にアクティブ化されることはなく、クロック5がアクティブ状態の時クロック4がアクティブ化されることもないように構成がなされている。

トークンラッチのロジックはメモリ10-15に付随するもののなかでどの叉点ロジックが最後にこのメモリにアクセスしたかを記録する。この記録は、その叉点ラッチが論理値「0」のB1信号を出力することにより表示される。このトークンラッチのロジックは、次に説明する優先化ブロックと連携して動作して、メモリに最後にアクセスした叉点に対し、その後そのメモリに対して同時アクセスが試みられた場合に最下位のアクセス優先権を付与する。トークンラッチの内容が変更される仕方については、次の優先化ブロックの説明の後に説明する。

優先化ブロック1902は2つの2入力NANDゲート1911及び1912を有する。ゲート1912の2つの入力ゲート1910及び1911の出力より供給される。ゲート1912の出力は下方の叉点(1-4)の信号Aに接続される信号A1である。ゲート1911の1つの出力は2番目に高い縦列叉点(叉点1-6)に対応する論理回路中のトークンラッチからの信号B1に接続される前述の信号Bである。もう1つの出力は2番目に高い縦列叉点(叉点ロジック1-6)の優先化ブロックからの信号A1に接続される信号Aである。(これについても前述)。

優先化ロジックは、メモリを最後にアクセスした叉点ロジックの下方のロジックを始点とする循環リップル状の経路を形成する。メモリを最後にアクセスした叉点ロジックはB1信号が「0」になることによって指示される。B1信号が「0」になると、ひとつ下方の縦列叉点のゲート1911の出力が論理値「1」になる。この出力は信号A1を得るためにゲート1912によってゲート1910の出力とゲートされる。ゲート1910の出力が「1」で、アドレス認識ロジックによるアドレス一致が検出されなかったことが指示されると、信号A1は「0」になる。この信号はひとつ下位の縦列叉点に伝送され、ゲート1911の出力を論理値「1」にすると共に上記循環リップル状経路のゲートの出力を次々に「1」にする。しかしながら、ゲート1910の出力が「0」であると、信号A1は次の叉点に論理値「1」として出力される。すると、後続のB入力がすべて「1」になっていることと合わせ(リップル状経路の始点だけが「0」のB信号を出力し得るから)、リップル状経路の周りの他のすべてのゲート1911は論理値「0」を出力する。このように、叉点はそのゲート1911の出力が「1」で、ゲート1910の出力が「0」

の時に限ってメモリにアクセスすることができる。この条件が満たされるのは、アドレス認識ブロックによりアドレス一致が検出され、その叉点が循環リップル状経路の始点から最初にメモリアccessを要求した叉点の場合のみである。

次に、トークンラッチの内容の管理について説明する。ゲート1913及び1914は、最後にメモリアccessした叉点は必ずトークンラッチ「0」を保持するようにするために設計されている。ここで、次のような事例を検討して見る。

① トークンラッチ1904のトークンが「0」で、どのバスもメモリアccessを必要としていない。この「0」は循環キャリア経路を完全にリップル状に一巡して出発点の叉点の信号Aに「0」として戻り、ゲート1911の出力を「1」にする。トークンラッチに既に保持されている「0」(信号B1)がゲート1913の出力を「1」にする。これら2つの信号はゲート1914の出力を「0」にし、この「0」がクロック4によってラッチ1945/1946にロードされ、こうしてトークンラッチに「0」を保持することによりリップルの伝播が持続される。

② トークンラッチ1904のトークンが「0」で、他の叉点の1つがメモリへのアクセスを要求している。この場合、信号Aが「1」として戻され、これが入力Bが「1」であることと合わせ、ゲート1911の出力を「0」にし、ゲート1914の出力を「1」にする。これはクロック4によってトークンラッチ1904に「1」としてロードされる。このトークンラッチは他の叉点の1つが丁度メモリアccessを果たしたために、「1」になる。

③ トークンラッチ1904のトークンが「1」で、より高い優先権を指定されたある叉点がメモリアccessを要求している。この場合、信号A及びBはいずれも「1」として受信され、上記の例同様、トークンにはやはり「1」がロードされる。

④ トークンラッチ1904のトークンが「1」で、その叉点がメモリアccessを要求しており、それより優先順位の高い叉点は全くメモリアccessを要求していない。この場合は、AまたはBは「0」として受信され、ゲート1911の出力は「0」になる。ゲート1910の出力は、アドレス認識ロジックがアドレス一致を検出しているので「0」となる。するとゲート1913の出力が「1」になる。ゲート1914の両入力は「1」であるから、「0」が出力され、これがクロック4によってトークンラッチ1904にロードされる。その結果、トークンラッチはメモリアccessをグラント(許可)された(グランテド)ばかりであるから内容が「0」になる。

ロジックの4番目のブロックはグラントラッチである。ゲート1910の出力はインバータ1940を介して2入力NANDゲート1915の一方の入力に供給される。ゲート1915の他方の入力はゲート1911に接続されている。ゲート1911の出力が論理値「1」でゲート1910の出力が「0」で

10

20

30

40

50

あるという条件はゲート1915の出力「0」にする。(それ以外の条件ではゲート1915の出力は「1」)。この条件は又点が首尾よくメモリアクセスを許可された時に満たされ、且つそのメモリに対応する又点の1つについてのみ起こり得る。ゲート1915の出力はクロック1によりトランジスタ1951を介してラッチ1941/1942にロードされる。(実際には、クロック1とクロック4は、トークンラッチとグラントラッチが同時に更新されるよう同時に作用する)。ゲート1942の出力はクロック2によりトランジスタ1952を介してラッチ1943/1944にロードされる。ゲート1944の出力は又点スイッチ1905への接続信号を発生するゲート1949を通して又点スイッチ1905はこの信号によってプロセッサバス1932をメモリバス1931に接続する。これらの又点スイッチは最も簡単なものでは個別のn型トランジスタで形成することができる。

ゲート1942の出力は、信号1934をトランジスタ1957のソースを接続するトランジスタ1958のゲートにも供給され、トランジスタ1957のドレインは接地されておりゲートはクロック2に接続されている。トランジスタ1957及び1958は、その又点が首尾よくメモリアクセスを許可された時、信号1934を接地に落とす。この状態はプロセッサにメモリアクセスを続行することができるということを指示する。しかしながら、信号1934はメモリアクセスが試みられている時はローにならず、他の又点がメモリアクセスを得、プロセッサはホールドし、メモリアクセスを再請求しなければならない。ここで説明したラウンドロビン優先化(優先順位指定)のやり方によれば、僅かな限定された回数のアクセスリトライを実行するだけで確実にメモリアクセスが許可されるようになっている。

第20図はクロスバ信号のタイミングを示す。この図では、PP2とPP3はどちらも各サイクル毎に同じRAMをアクセスしようとするが、ラウンドロビン優先順位指定方式のためこのアクセスは交互に行われる。PP2はアドレスS、T及びUを計算して出力し、PP3はアドレスV及びWを計算して出力する。「5 MSアドレス」信号から、グラント信号をどのように用いて最後のアドレス(リトライの場合)と計算中の新しいアドレスを多重化するかは明らかであろう。パラレルプロセッサは、グラント信号がスレープフェーズの終わりまでにアクティブ化されないと、コンテンションが起こったと見なし、フェッチ、アドレス及び実行の各パイプライン段階のマスター更新段階はキル(抹消)される。

スイッチマトリクスの集積化

前にも述べたように、このシステムでは、メモリコンテンションは各又点に各個に設けられた論理回路群を有するトークンパッシング方式により処理される。この発明の一実施例では、この論理回路群は各又点に直接付属させて配設されている。このように、又点は各々のポートと共にシリコン基板上に空間的に分散配置されている

から、コンテンション制御ロジックも同様に空間的に分散配置されている。これによれば、スペース節約が可能で、実際の回路のロジックもクロスバスイッチの容量と共に大きくすることができる。このようにして、余分のシリコンチップスペースを費消することがないように、ロジックをシリコン基板の複数層の中の1層中に配設することができる。これにはスペース節約の他、トークンパッシング回路に対する配線量を最小限に抑え得るという長所もある。

10 同期式MIMD

第21図に示すように、プロセッサ100~103には各々に対応させてレジスタ2100~2103が設けられており、これによって同期動作が要求されているかどうかが表示される。また、以下に述べるように、各プロセッサと同期された他のプロセッサのアドレス(アイデンティティ)を保持するためのレジスタもそれぞれ設けられている。命令ストリームには、これらのプロセッサと同期して実行しなければならない一連の命令の始めと終わりを示す命令が含まれる。同期された命令ストリームを開始させるコードがあるプロセッサに伝達されると、そのプロセッサ及びこれと同期された一群のすべてのプロセッサは、同期されたコード命令の終わりが検出されるまでは、互いにロックされたステップ関係(同期関係)でしか命令を実行することはできない。

この手法によると、プロセッサ間でメッセージを転送する必要が全くなく、それらのプロセッサは実行中の命令ストリーム次第で1サイクルの間または何サイクルかの間インステップ(同期)状態を保つ。各プロセッサ間に同期関係を確立するのに命令ストリーム以外の外部制御は全く不要である。

次に、第22図に移って、パラレルプロセッサ100~103には、4ビットの同期レジスタ2207が内蔵されており、その4ビットにはプロセッサ103、102、101及び100に対応させて3、2、1、0のラベルが付してある。各レジスタの各ビットがそれぞれ1つのプロセッサに対応している。ある特定のプロセッサを同期させるその他のプロセッサはそれらのプロセッサに対応するビット位置に「1」を記入して指示してある。また、互いに同期関係になるはずの他方のプロセッサにあっては、各々の同期レジスタの適宜のビットが同じくセットされている。

同期して実行することが望ましいコードは、LCK(ロック)及びULCK(アンロック)命令でバウンドすることにより指示されている。LCKの下側に記されているULCKまでの命令は他のパラレルプロセッサとのロックされたステップ関係の下にフェッチされる。(従って、同期された各パラレルプロセッサではLCKとULCK命令の間には同数の命令がなければならない。)

ここで望ましいのは命令の同期フェッチングよりむしろデータ転送を通常よりも多く同期化することである。しかしながら、LCK命令及びULCK命令の直前までの命令

(直前の命令を含む)と並列にコード化された転送が同期しておこなわれるというのはパラレルプロセッサのパイプラインの故である。このような転送は、必ずしも正確に同一マシンサイクル内で行われなくともよいが(メモリアクセス・コンフリクトのため)、次の命令でコード化された転送は、前の命令の同期された転送がすべて行われるまでは開始されない。そうでないと、ロードや記憶の順序がメモリアクセスコンフリクトによって混乱してしまう。

同期したコードが実行されているという情報は、各状態レジスタ(ステータスレジスタ)のS(synchronize;同期された)ビットにより記録される。(このビットは、LCKまたはULCK命令のアドレスパイプライン段階のマスターフェーズまで実際にセットあるいはリセットされることはないが、LCKまたはULCK命令の実行はスレーブフェーズにおける次の命令のフェッチに影響を及ぼす。このビットはリセットによりクリアされ、また状態レジスタがプッシュされていると割込みによってクリアされる。

やはり第22図において、各同期レジスタ2207の4つのビットは色々なプロセッサの間の所望の同期の仕方によりソフトウェアによってセットされる。そのため、プロセッサ100をプロセッサ103と同期させるものとする、図示のようなビットが各レジスタ2207にロードされる。例えば、1、0、0、1という内容はプロセッサ3をプロセッサ0に同期させるということを示す。また、図示例においては、プロセッサ101と102を互いに同期させるため、各々の同期制御レジスタにはビット0、1、1、0がロードされている。

次に、プロセッサ100について見ると、同期レジスタ2207のいずれかのビットに「0」があると、それらのビットが入力されている各NANDゲートの出力に論理値「1」が現れる。図示の例では、NANDゲート2203及び2204の出力がそれぞれ論理値「1」になる。これらの「1」はNANDゲート2206の入力に供給される。NANDゲート2206はその入力がすべて「1」になるまでは、プロセッサ100に如何なる命令も実行させない。図示の場合は、レジスタ2207の1及び2のビット位置に「0」があるため、ゲート2203及び2204はバス40のリード1及び2上の信号を無視する。従って、この場合、コードの実行は、バス40のリード0及び3上の情報に反応してゲート2206により制御されることになる。ロック命令が発生するとSビットがセットされ、ゲート2201の一方の入力が論理値「1」になる。ここでは、一寸の間プロセッサにおける実際の実行タイミングを制御する信号であるOK T OSYNC信号の存在を無視することにする。ゲート2201の出力は各プロセッサの同期レジスタ毎にそれぞれ異なるリードに接続されている。例えば、プロセッサ100のゲート2201の出力はリード0に接続されているが、プロセッサ101のゲート2201の出力はリード1に接続されてい

る。ここで重要なのは、プロセッサ100のゲート2201の出力はその同期レジスタ及び他のすべてのプロセッサの同期レジスタにおけるゲート2205の0入力(バス40のリード0に接続されている)に接続されているということである。この点は、プロセッサ101及び102の場合、ゲート2205の一方の入力が各々の同期レジスタの論理値「0」に接続されているので、何ら影響がない。しかしながら、ゲート2205の一方の入力がレジスタの論理値「1」に接続されたプロセッサ103においては、バス40のリード0上の出力、従って実際にはプロセッサ100のゲート2201の出力によって制御されることになる。このようにして、プロセッサ103はプロセッサ100内で生じる動作により制御されるが、ここでプロセッサ103をプロセッサ100に同期させようとしたのであれば、これはまさしく所期の結果である。図示の回路群を見れば、プロセッサ103のゲート2201の出力はバス40のリード3に接続されており、このリード3はプロセッサ100のゲート2202の入力に接続され、ゲート2202は同期レジスタ2207から他方の入力に供給される「1」によっても制御されるので、逆にプロセッサ103からプロセッサ100へも上記と同じ動作が行われるということは明らかであろう。

ゲート2201に入力されるOK TO SYNC信号については、この信号が論理値「1」になると、コードの実行が許可され、同期レジスタのビット0の位置に「1」が入っている他のプロセッサがすべてその信号と同期して動作する。このようにして、OK TO SYNC信号がローになってキャッシュメモリあるいはコードの実行に関連して何らかの問題があることが指示されると、この信号と同期したすべてのプロセッサはその問題がクリアされるまで待機状態となる。以上の如く、プロセッサ間には各レジスタに定期的に記憶されるコードにより制御されるところに従い完全な同期関係が確立される。この場合、全部のプロセッサまたはそのいずれかの組合せについて互いに同期させることもできれば、プロセッサ間で何種類でも異なる同期関係が用いられるようにしてもよい。

同期されるのは命令フェッチであるから、同期化されたコードに割り込みを掛けることも可能である。このような割り込みはすぐにパラレルプロセッサの同期信号をイナクティブにする。キャッシュミス及びコンテンションはほぼ類似の結果をもたらす、ハードウェアをインステップ状態に保持する。ただし、コンテンションの場合は、コンテンションが生じた命令に続く2つの命令がパイプライン中にフェッチされてからパイプラインが休止する。

アイドル命令を同期されたコードに入れて、あるパラレルプロセッサが割り込みされた後その割り込みルーティンから復帰するまで、同期されたすべてのパラレルプロセッサのオペレーションを休止させることも可能である。

同期されたコードに割り込みを掛けることができるようにする必要があるのであるため、いずれか1つのプロセッサのプ

ログラムカウンタPCを行先として指定した命令は、すべて状態レジスタのSビットの作用を直ちにディスエーブル化（無効化）するが（ULCKT命令と同じタイミングで）、Sビットはセット状態を保持する。いったん2つの遅延スロット命令が完了すると、Sビットの作用が再びイネーブル化（有効化）される。このメカニズムによれば同期された遅延スロット命令に割込みが掛けられなくても何ら問題はない。そのために、同期ロジックはブランチ（分岐）、コール（呼出し）及びリターン（復帰）を単一の命令として取り扱う（2つの遅延スロット命令が続くPCロードとして実行される）。同期信号は2つの遅延スロット命令の間イナクティブ化され、これらの命令は同期信号を見ることなくフェッチされる。LCK命令が遅延スロットに入れられた場合、その効果は遅延スロット命令が実行された後に発揮される。同期されたループは、そのブランチは実行段階ではなくフェッチパイプライン段階で働くので、通常のコード同様の挙動を示す。

このシステムにおける同期の作用の仕方の一例が第23図に示されている。この場合、パラレルプロセッサ2とパラレルプロセッサ1は、A0とA1が各パラレルプロセッサに同じアドレスを有していると仮定すると、各々のデータD0レジスタ（第33図参照）の内容を交換する。ここでは、また、A0とA1はコンテンションを避けるために互いに異なるRAMを指示するものと仮定する。（同じRAMを指示しても有効に作動するが、余分のサイクルが必要である。）

この例では、パラレルプロセッサ1は、パラレルプロセッサ2がそのLCK命令に達してから1サイクル後にLCK命令に達する。このように、パラレルプロセッサ2は1サイクル待機する。そして、これらのパラレルプロセッサは同時に記憶を行うが、パラレルプロセッサ2は、ロード命令をフェッチする時キャッシュミスを起こす。両パラレルプロセッサはこのキャッシュミスが転送プロセッサによって処理されるまで待機する。その後、これらのプロセッサは同時に且つ同じようにULCKをロードする。次いでパラレルプロセッサ1が命令4をフェッチする時にキャッシュミスを起こすが、これらのプロセッサは今度はアンロックされているので、パラレルプロセッサ2は妨害されることなく動作し続ける。

SIMDモードでの同期は、その方式自体がもともと同期性を含んでおり、LCK及びULCK命令が目的を持たず、従ってコード化されたとしても何ら作用しない。状態レジスタ（ステータスレジスタ）のSビットは、たとえば「1」にセットされているとしても何の効果もない。

命令（LCK）はMIMDの同期されたパラレルプロセッサコードを開始するのに用いられる。この命令は、そのパラレルプロセッサを同期レジスタの「1」により指示されたパラレルプロセッサがすべて互いに同期するまで待機させる。そして、次の命令がその他のMIMDパラレル

ロセッサと同期してフェッチされる。アドレス及び実行パイプライン段階の実行は、連続する各命令が同期してフェッチされる毎に行われる。この命令のアドレスパイプライン段階の間状態レジスタのSビットはセットされている。

命令（ULCK）はMIMDのパラレルプロセッサを互いにアンロック状態にする。すると、これらのパラレルプロセッサは次の命令フェッチ時に独立の命令実行を再開する。

10 スライス式アドレス指定

スライス式アドレス指定は、1つのメモリ空間により隣接情報を取り出し、その情報を配分後に、多数のプロセッサがコンテンションを起こすことなく同時にアクセスすることができるようなやり方で多数の別個のメモリ空間に配分する技法である。

例えば、第24図には0～127と番号が付された1行の隣接ピクセルを有する外部イメージメモリバッファが示されており、この行には「a」と言う記号が付されている。この情報はスライスアドレス指定技法を用いてバス2401を介しメモリサブシステム10に伝送され、最初の16のピクセル（0～15）はアドレス0～15を割り当てられた第1のメモリ10-0に入れられる。その次の16のピクセルはメモリ10-1に入れられる。この例では、上記のプロセスが8つのメモリについて連続して行われ、最後のメモリ10-7にはピクセル112～127が入れられる。スライスアドレスロジック2401は転送プロセッサで実行されると共に、以下に説明するように、パラレルプロセッサのクロスバアドレスユニットでも実行される。

上記の例の場合、従来技術のアドレス計算手段では128の連続アドレスが生成されるはずである。これはデータが1つのメモリ内に入れられるということを意味する。この例では、データはメモリ10-0の連続したアドレスに現れる。従って、複数のプロセッサによるこれらの情報への同時アクセスは、これらのプロセッサがいつべんに同じメモリをアクセスしようとするため、不可能である。そのため、従来技術では、ピクセル0～15がメモリ0の行Aに入り、ピクセル16～31が行Bに入り、行Cにはピクセル32～47が入るというような動作が127の隣接ピクセルがすべてメモリ0の色々な行に入るまで繰り返される。このように、従来技術にあつては、異なるプロセッサが並列に動作して情報を処理するが、これらのプロセッサがすべて色々なピクセルバイトを求めてメモリ0を競合してアクセスするので、時間が無駄に消費され、パラレルプロセッサの価値が減殺される。

第25図はいろいろなビットのアドレスの場所を制御するのに用いられる従来技術の加算器を示し、この図には、3つのシングルビット加算器2501、2502、2503が描かれている。これらの加算器はメモリのアドレス範囲に等しい数のシングルビットを有する全加算器の一部である。これらの加算器は、アドレスの1ビットが各加算器

2501～2503の各A入力に供給されるように動作する。即ち、アドレスの最下位ビットは加算器2501に入り、最高位ビットは最高位のシングルビット加算器2503に入る。

B入力には記憶のためにアドレスに指標付けされる量の2進数字が入る。加算器2501～2503の組合せによって得られる結果アドレスはメモリアクセスのために用いられる。各加算器は1つ高位の加算器のキャリー信号入りにキャリー信号を出力する。各加算器のビット入力には3つの入力A、B及びキャリーインが取り込まれ、これらの入力のいずれかに2つ又は3つの「1」があると、そのセルからのキャリーアウトは「1」になる。このキャリーアウトは加算器の1つ高位のキャリーイン入力に供給される。このプロセスが各個別の加算器ビットについて繰り返され、メモリ空間をアクセスするのに必要なサイズの結果アドレスが生成される。各キャリーアウトが1つ高位のキャリーインに直接接続されているということは、結果アドレスは常に隣接アドレス空間の一部になっているということを意味する。前出の例の場合、値「1」のインデックスが加算器のB入力に供給されるとすると、メモリへの結果アドレス出力はA入力に供給された最初のアドレスより「1」だけ大きくなる。

次に、第26図には上述の通常に加算器を改良したものが示されている。図示の改良した加算器によれば、各セルのキャリーアウトは各セルに供給されるキャリーイン信号と多重化され、これによって加算器の1つ高位のキャリーイン入力に伝送される信号を前のセルのキャリーアウトかまたはそのキャリーインのいずれかに選ぶことができるようになっている。例えばセル2505について見ると、そのキャリーアウト信号はマルチプレクサ2508に供給され、このマルチプレクサの他方の入力にはセル2505へのキャリーイン信号が供給されている。信号Bはマルチプレクサ2508を制御して、る2505のキャリーアウト1またはキャリーインのどちらかを高位側の次のセルのキャリーイン入力に通過させるようにするために用いられる。

前述の標準加算器セルのもう1つの改良として、このシステムでは、マルチプレクサの信号Bを制御すると同じ制御信号を供給されるADDというラベルを付した制御入力を設けている。この場合、論理値「1」が信号B上に供給されると、セル2505のキャリーイン信号が次の高位セルのキャリーイン信号に供給される。また、信号B上に論理値「1」があると、入力Aに供給された最初のアдресがそのままストレートに出力に通されるようにセル2505の加算機能が抑止される。これは入力B上の「1」の存在に対応するアドレスビットを保護する効果がある。ここで、この改良された加算器の制御信号に多くの「1」を供給することによって、最下位ビットのセルからのキャリーアウトは、加算器の全長に及ぶ沢山のセルに伝播した後、あるセルのキャリーインに供給され、加算機能を実行させることができるということは明

らかであろう。このセルは、ADD制御信号上に「0」がある高位側の次のセルである。この効果は、アドレス範囲の多数のビットが変更されないようにして、バイパスされたセル内に書き込まれているアドレスを保護することにある。前述の例では、「1」をマルチプレクサ及び各セルのADD制御信号上に供給することにより、メモリ0中のピクセル15からメモリ1のピクセル16へアドレスインクリメントを行い、これによってメモリを1つの連続アドレス空間としてアドレス指定することができる。マルチプレクサ制御信号は、一部のビットをアドレス範囲からマスクアウトし、第24図に示すようにメモリに分散されたデータをスライスとしてアクセスさせるので、スライドマスク信号とよばれる。

ここで、この回路群は隣接情報を記憶するためばかりでなく、隣接情報を検索するためにも用いられるということに注意すべきである。また、一部の情報は同じメモリに供給され、記憶されるべきであり、スライスされてはならないが、このことはスライスマスクのABCリードに全て「0」を供給することにより指示される。この条件下にあっては、加算器2502～2506は各々従来技術の加算器2501～2503と全く同様に動作する。また、いくつかのメモリに渡ってスライスされるべき分散形データはピクセル情報ばかりでなく、何種類か型があるということも重要である。これはいくつかのプロセッサがその時点でどのような処理が実行されているかに関わらずその処理のために同時に同じ型の情報をアクセスするようなことが想像されるばかりには常に重要である。

また、スライスアドレス指定方式のところで開示したようにしてメモリの分散を行うと、特定のピクセルあるいはその他のデータの記憶には使用されない行B及びCがその他の情報ように使用されるので、メモリの浪費は全く生じないということも重要である。唯一起こり得ると考えられる不利は、マルチプレクサ及び加算器の相互接続配線の付加部分を組み込むのに余分のチップスペースが必要ということである。この不利は、並列処理におけるメモリアクセスの速度が劇的に速くなり、しかも隣接情報を多くのメモリに渡って分散させる場合及び外部制御による制御下において単一メモリに情報を記憶させる場合の両方の場合について十分なフレキシビリティを確保し得ると言う成果に比べれば取るに足りないものである。この手法を用いる場合、何らかの特定の情報との固定された関係は全くないので、情報の用途に応じて、色々な時点で多くのメモリに情報を分散させることもできるし、あるいは同じ情報を異なる時点において同じメモリに記憶することもできる。

例えば、並列処理モードで使用するために一度スライスされた情報を後である期間一回だけ単一プロセッサ用に用いることが決まったような場合は、その期間についてはスライスマスクに全て「0」を入れることによりその情報を単一メモリに記憶して単一のプロセッサがその

単一メモリにアクセスすることができるようにする方が有利であると考えられるし、これによってもスライス方式に増してさらに貴重な時間節約を図ることができる。これはシステムの設計並びにデータ記憶のためのオペレーションモードに対し高度のフレキシビリティを与えるものである。

次に、第27図には、典型的な量のピクセルを多数のメモリ上に分散させるやり方の一例が示されている。図示の場合、各メモリの容量は2キロバイトであり、これらの各メモリのスタートアドレス（開始アドレス）及びエンドアドレス（終了アドレス）が示されている。例えば、メモリ0のスタートアドレスは0000でありエンドアドレスは07FFである。メモリ1のスタートアドレスは0800であり、エンドアドレスは0FFFであり、以下同様にして、メモリ7ではスタートアドレスが3800であり、エンドアドレスが3FFFとなる。図示のように、これら複数個のメモリの間に多量のピクセルが1メモリ当たり64ピクセルずつスライスされて分散されている。ここでメモリ3内における64個のピクセルのステップ動作について一寸考察してみる。これらのピクセルは図示のようにアドレス1900～193Fの所に記憶されている。これに隣接する1単位の情報は、情報全体がスライス方式によりメモリアクセスに分散されているため隣のアドレス1940の所には記憶されていない。このことはその隣接の1単位の情報は、メモリ4のアドレス2100にあると言うことを意味する。従来技術による加算方法では、第27図に示すように、アドレス193Fに「1」の指標を加算してアドレス1940を作る。前述のように、このアドレスにあるのは必要な次の1単位の情報ではなく、その情報は次のメモリのアドレス2100にある。第27図の最下部に、スライス式演算による加算の仕方が示してあり、その中でアドレス値193が2進形式で示され、その下にスライスマスク情報も同様に2進形式で示されている。前にも述べたように、スライスマスク内に「1」があると、ある加算器セルからのキャリーアウトはキャリーバスに沿って高位側の次のセルより遠くまで伝達される。この例では、スライスマスクに連続して5つの「1」があるので、5つの加算器セルはキャリー信号によってバイパスされる。このように、ここで説明した改良形の加算器のB入力に供給された「1」の指標がその改良形のA入力に供給されるアドレス193Fの値に加算されると、下位側から6番目のビットからのキャリーアウトは7番目乃至11番目のビットをバイパスし、12番目のビットのキャリーイン入力に伝達される。これによって、下位側から12番目及びそれ以後のビットを含むアドレスのビットをインクリメントさせることができ、その結果、各メモリの容量は2キロバイトであるから、必要な次のメモリのアドレスまでインクリメントすることができる。

構成変更可能型メモリ

ここで、MIMD/SIMDオペレーションモードでどのよう

にしてメモリの構成変更が行われるかを説明する前に、前出の第4図によりプロセッサのメモリとクロスバスイッチの相互接続について簡単に見直しておく方が良いと思われる。前にも説明したように、MIMDモードでは、各プロセッサはそれぞれ別個のメモリから命令を得る。そのため、第4図の実施例においては、プロセッサ100はその命令ポートに接続された縦列リンク（命令縦列リンク）から又点19-7を介して命令メモリ10-1に接続されている。又点19-7は、転送プロセッサが命令メモリにアクセスしている時以外は通常閉じられており、転送プロセッサが命令メモリにアクセス中は又点19-7に信号が供給されてこれを制御すると共にオフにする。

同様にして、プロセッサ101はその命令縦列リンク及び又点14-7を介して命令メモリ10-5に接続されている。また、プロセッサ102はその命令縦列リンクから又点9-7を介して命令メモリ10-9に接続され、他方プロセッサ103は命令縦列リンクから又点4-7を介して命令メモリ10-13に接続されている。これはシステムがMIMDオペレーションモードにある時のメモリープロセッサ構成方式である。

システムの全部または一部がSIMDオペレーションモードに切り換えられた時は、SIMD及びMIMDモードが両方とも稼働するのか、あるいはSIMDモードは一部のプロセッサ群についてのみ稼働するのかによって、メモリ10-1を2つ以上のプロセッサに接続するかまたは一群のプロセッサに接続するようにする方が望ましい。図示の実施例においては、SIMDオペレーションは4つのプロセッサ100～103の全部について行われるものと仮定する。この場合、命令メモリ10-1は、又点19-7を介してプロセッサ100に接続され、またスリーステートバッファ403が又点14-7と共にアクティブ化されてメモリ10-1をプロセッサ101の命令縦列リンクに直接接続させるようになっている。同様にして、スリーステートバッファ402及び401が作動すると、命令メモリ10-1がそれぞれ又点9-7及び4-7を介してプロセッサ102及び103の各命令縦列リンクに接続される。

この時点で、プロセッサ100～103がすべてメモリ10-1より供給される単一の命令ストリームで動作するシステムが構築されたことになる。この場合、MIMDモードでは命令記憶用に使われるメモリ10-5、10-9及び10-13は他の目的用に自由に用いることができる。メモリ容量を少なくとも一時的に増やすため、これらのメモリはすべてのプロセッサがアクセス可能となる。以下、これについて詳細に説明する。

第28図において、レジスタ2820は現在進行中のシステムのオペレーションモードを示すデータを記憶する。即ち、このレジスタはシステムがMIMDモード、SIMDモードあるいはSIMDとMIMDの組合せ（ハイブリッド）モードの中のどのモードになっているかを示すビットを有する。このレジスタからはMIMDを示す信号とSIMDを示す信号の

10

20

30

40

50

2つの信号が供給される。図示実施例は、一対の信号しか示されていないが、実際は各プロセッサ毎に一対の信号が供給されるようにすることができる。これらの信号は又点及びスリーステートバッファに送られて、適切な構成のための適切な命令ストリームが選択されるようになっている。MIMDの構成においては、プロセッサ101、102及び103は各々独自の命令ストリームを実行する。これらの命令ストリームはプログラムカウンタ2811、2812及び2813によってそれぞれ指示される。これらのプログラムカウンタの内容はそれぞれキャッシュロジック2810、2802及び2803に供給される。これは、プログラムカウンタにより指示された命令が現在それぞれメモリモジュール10-5、10-9及び10-13にあるか否かを指示する作用を有する。プログラムカウンタにより指示された命令がこれらのメモリモジュールにあれば、MIMD命令アドレスがキャッシュロジックから各メモリへ出力され、そのメモリから逆に命令縦列リンクを介して各プロセッサへ適宜の命令ストリームがフェッチされる。この時、メモリ内に命令がなければ、命令実行は停止され、又点13-0、8-0または3-0（第4図参照）が転送プロセッサのバスに接続され得る状態になる。これらの又点は、各プロセッサによって、実行する必要のある命令の外部アドレスを伝送したり、また次の命令ストリームが記憶されるはずの命令メモリ10-5、10-9または10-13内の記憶場所を伝送するのに用いられる。転送プロセッサがいったんこれらの命令をフェッチすると、転送プロセッサよりコードがフェッチされたことを示すアクノリッジ信号（応答信号ACK）がパラレルプロセッサへ送られる。すると、パラレルプロセッサは、例えば命令ストリームがないことが発見されて上記のプロセスが再度リピートされるような時点まで、やはり命令メモリからの命令を実行することができる。

SIMDモードの構成では、プロセッサ101、102及び103は同じ命令ストリームを実行するので、プロセッサ内のキャッシュロジック2801、2802及び2803は何の機能も果たさないことによりディスエーブル化される。プログラムカウンタ2811、2812及び2813の内容は、SIMDの構成では命令はすべてプロセッサ100によりフェッチされ、これらの内容は命令フェッチにおいて何の機能も果たさないため無関係である。したがって、SIMDの構成においては、メモリ10-5、10-9及び10-13はデータ記憶用として用いることが望ましい。そのためには、又点14-6、9-1乃至9-6及び4-1乃至4-6がイネーブル化され、これによってこれらのメモリのデータがプロセッサによりアクセスすることができる状態にする。このことは、システムにおけるメモリの利用がSIMD及びMIMD構成の両方について最適レベルに保たれるということの意味するものである。

画像処理用パーソナルコンピュータ

第46図乃至第52図は、画像処理用のパーソナルコンピ

ュータ（PC）を示すが、図示のパーソナルコンピュータは3つの主要構成要素、即ち第46図に示すカメラセンシングデバイス4600、画像処理デバイス4602及び第48図に示すようなディスプレイデバイス4801で構成することができる。この画像処理用PCは必ずしもカメラ4600やディスプレイ4801を使用することに限定されるものではなく、多くの様々な形態の入出力手段を用いることができる。

カメラ4600は例えばPCの前面4601に焦点を当て、手4603を用いてよく壺嚙者との意志伝達で行われるように「手合図」をすることにより情報を入力することができる。この「手合図」はカメラによって観ることができるし、またスクリーンを用いて「指2本」のサインを表示することもでき、あるいは第11図によって前にも説明したようにして画像情報をさらに処理することも可能である。パーソナルコンピュータからの出力バスは、カメラ4600より入力される情報のデジタル表示をも含めて伝送するようにしてもよく、この場合のデジタル信号としては2進信号が用いられる。従って、ユーザはスプレッドシートやその他の情報取得手段、例えばASCIIコードによりキーボードまたはその他従来の方式で情報を取得する手段並びにカメラ4600またはビデオレコーダあるいは画像処理コードを用いるその他の形態のビデオ入力のようなビジュアルソースまたはビデオソースから情報を取得する手段等を使用することができる。ビデオ入力はテープ、ディスクまたはその他のメディアに記録することができると共に、PCへ供給するために情報を記憶するのと同じやり方で記憶することができる。

このような画像処理用PCは例えば下記のような特徴を具備し得る：1) カメラ、スキャナ及びその他のセンサよりイメージを取得する；2) 文書（ドキュメント）中の情報あるいは物を理解する；3) 文書または画像から関連情報を抽出する；4) 画像や文章ドキュメントを組合せたデータベースを通じて問題を切り抜ける；5) ジェスチャ認識のような進んだ画像処理用インターフェースを提供する。

このPCは、このシステムに入れられた情報は読み出すことができ、また情報内容は他のシステムによってさらに処理することなく直ちにアブストラクト化できるので、インスタントデータベースを創出するのに使用することができる。これによれば、記憶前にいずれも全く認識されなかつた特定語（ワード）の一致により簡単にアクセスすることが可能なデータベースが創出される。これは、そのような特定語だけでなく幾何学形状やピクチャにまで拡張することができ、多くの用途に効果的に利用することが可能である。例えば、カタログや新聞を走査して、ハイウェイ上のすべての樹木、すべての赤い車あるいは一定寸法を超える全てのトラック等、特定の対象を探するためのシステムを設計することもできる。そして、概念的には、イメージプロセッサがアブストラクト

してユーザが使い易くしたワード、物、及び形状によりデータベースが形成されることになる。

このような画像処理能力を有するPCの1つの用途は、静止画でも動画でもまたビデオでも、単にPCに画像を走査させるだけでシステム内に取り込むことができ、あるいは何らかのドキュメント中に組み入れることができるというような用い方である。このように取り込んだ情報は、次に、第11図により前に説明したように、アブストラクト処理され、その出力はユーザ制御下における以後の処理のために画像処理用PCで利用することができる。

この発明のシステムで何故そのように大きな画像処理能力が得られるかという理由の1つは、単一チップに多数のメモリと共に互いに並列に動作する複数のプロセッサが内蔵されており、しかもこれらのメモリがすべてシステムのほぼ瞬間的な再構成を可能にするクロスバスイッチによってアクセス可能であることによる。これによれば、従来では未知の高度の能力とフレキシビリティが確保される。このことは、また、画像処理能力の大幅な増強を可能にし、そのような大きな画像処理能力をその他の処理能力と共に利用することによりこれまで知られていなかった形のサービスを提供することが可能になる。その例としては、例えば、写真及びその他の画像の復元、修復、あるいはファクシミリ文書をそのバックグラウンドにある異質なものを取り除いて送信されたイメージと同程度あるいはそれ以上に鮮明にするクリーニング処理等が考えられる。このシステムは主に処理能力を1つのオペレーション単位に組み入れられる故に、比較的小さなパッケージに封入することができる。バンド幅に関する制約や配線等の物理的制約の問題も解消される。

この構想を拡張すると、画像処理PCを、腕に装着可能で大きなビデオディスプレイの代わりに小さなフラットパネルを用いた小さなユニットに組み込み、第46図に示すように、例えばユーザが上方で指を振り、その像を入力するというようなことが可能となる。この画像処理システムは、前述したように、種々の動き（運動）を認識し、それらの運動を入力に変換（翻訳）する。これによれば、キーボードやその他の機械的入力デバイスの問題が効果的に取り除かれ、これらの代わりにビジュアルイメージを入力として用いることができるようになる。この場合入力デバイスはディスプレイともなり得、二重の目的に使うことが可能である。すると、このことは、光学式文字認識装置は現在考えられているよりもずっと重要な入力手段になるということを意味する。

第47図は、画像処理PCによる制御並びに第1図及び第2図の構成により実行される第11図のアルゴリズムの下に2本の指の像から求められた2進出力を示す。

また、第48図は画像処理PCを用いたリモート伝送システムを示す。

第48図乃至第52図は、イメージシステムプロセッサのPCを様々な用途に応用した実施例を示している。例え

ば、第49図は多重化された入出力デバイスを有するパーソナル用のデスクトップ型画像処理PCを示し、この例では、物体またはコピー対象書類は、光学系4907及びCCD（電荷結合素子）装置によって撮像あるいは検出される。ここで検出された情報は、A/Dデータ収集装置4904によってアナログ情報からデジタル情報に変換される。このデータ収集装置4904は、デジタルの検出情報をイメージシステムプロセッサ（ISP）に供給する。

10 コントローラエンジン4905は、CCD装置及びプリントアセンブリ4909の双方に所要のタイミング信号を供給する。プリントアセンブリ4909は文書（コピーまたはファクシミリプロダクト）4910を出力する。制御コンソール4902は、例えばキーボード、マウスあるいはその他前述のイメージ入力用デバイスで構成することができる。LCD（液晶）またはCRT（ブラウン管）ディスプレイ4903は例えばユーザへのビジュアル情報提供のために用いられる。ディスプレイ4903、ISP/メモリ4900及びプリントアセンブリ4909は、処理された画像データを伝送する画像情報バスにより互いに接続されている。

20 第50図はイメージシステムプロセッサ5000の隠れた応用例を示し、この例においては画像はやはり保安施設における侵入者の有無など外界から情報を収集するCCD装置5004またはその他のセンサによって検出される。この情報はイメージシステムプロセッサ5000の外部メモリであるフレームバッファまたはVRAM5003に記憶される。あるいは、このISPをパターン（または人物）認識装置として用い、出力制御情報をラッチ5009に供給するようにしてもよい。この情報は例えばドアロック、工場における工程等のような被制御機構5009を制御するのに用いられる。また、ラッチ5008からの出力は出力ディスプレイ5010にも供給することができる。プログラムあるいは命令はあらかじめハードディスクドライブ5002または光ディスク5001に記憶されている。これらのデバイスは上記の保安施設等における侵入者のイメージなど、何らかの情報の発生を記憶するのににも使用することができる。統計学的な記録蓄積手段5007では、システムの状態及び何らかの事象の発生に関するデータが記録され、蓄積される。

40 第51図はハンドヘルド型の画像処理用PCの概略構成を示す。この場合、イメージシステムプロセッサ5106は位置情報入力を供給する2つのCCDデバイス5105より入力を得、この位置情報入力を処理してユーザが与えたジェスチャの情報及びPCの制御情報を抽出する。そして、例えばユーザの手または疑似ペンの位置と方向がデバイスの制御のために、あるいはISPと共に有意のメッセージまたは文字を抽出するために用いられる。フラットパネルディスプレイ5104はこのハンドヘルド型PCの出力情報を表示する。オプション的には、外部カメラ5103を用いれば、ユーザは例えばハンドヘルド型PCの視野外のイメージを収集することができる。また、ホストポートある

いはプリンタポートを設けるならば、ユーザはハンドヘルドPC内の情報をダウンロードし、あるいはプリントすることができる。

第52図は、ホスト5205を有するネットワーク構成におけるISPの応用例を示し、この構成では、リモート方式であるいは何らかの中央局でオフライン収集されバッファ5201に配分された所要画像情報がホスト5205により供給され、図示構成の画像処理用PCはこの情報を用いてイメージシステムプロセッサ5200に情報を供給する。情報を取得する1つの方法は、フロントエンドプロセッサ5206と共に用いられるスキャナ5207によるものである。このような画像処理用PCの廉価版（第49図の構成に比べて）によれば、イメージ収集デバイスのネットワーク化によって資源共用（リソースシェアリング）が可能となる。また、プリンタインターフェース5203及びこれに接続されるプリンタ機構5204によりプリンタポートを得ることもできるが、これによれば、ユーザは画像あるいはイメージシステムプロセッサによりエンハンスされたイメージの外、普通の文章情報及びグラフィック情報を含む複合文書をプリントすることができる。

並列処理とメモリアクセスがすべて単一チップ上で行われ、かつこれと相俟ってオールチップ制御になるプロセッサメモリ構成とオペレーションモードの広範なフレキシビリティが確保されるこの画像処理システムのコンパクトな構成は、画像データ入力並びにASCII入力を受け入れる画像処理用PCの能力及びこれら2種類のデータを同時に利用する能力を助長するものである。

「1」計数回路

第53図は色々な処理アルゴリズムを用いて画像データを処理する動作が可能な画像処理システム5310を示す。ビデオカメラ、スチールイメージカメラ、バーコードリーダー等の画像処理用デバイス5312（画像用デバイス）は、画像を捕捉するために用いられ、そのデータを画像データメモリ5314に供給する。このように捕捉された画像のデータは、アドレスジェネレータ5318によりアドレス指定されたイメージプロセッサ5316によってアクセスされるまで画像データメモリ5314に記憶される。第1図及び第2図に示すプロセッサのようなイメージプロセッサ5316は、ヒストグラム作成等画像データに対する統計学的プロセスを含む信号処理動作を実行する。また、画像データ中の「1」の計数値を得るために「1」計数回路5320が設けられている。画像データ中の投影線沿いの「1」の数のような情報は画像データの統計学的分析のために用いられ、その分析結果はパターン認識等に用いることができる。例えば画像データのヒストグラムを所定の画像パターンのものと比較してそれらのパターンが一致するかどうかの認識を行う。出力デバイス5322はイメージプロセッサ5316に接続されており、画像処理システム5310の何らかの出力を表示するのに用いることがで

きる。出力デバイス5322としては、モニタテレビあるいはハードコピー作成デバイス等を用いることができる。

上記の画像処理システム5310について少し考えて見るならば、この発明を有利に適用し得ると考えられる環境の一例がわかるし、また上記の説明は何ら「1」計数回路の応用性を限定するものではないということは明らかであろう。

次に、第54図には「1」計数回路5320の論理ゲートレベルの実施例が示されている。図示の「1」計数回路5320は、カウントセル5426a～5426lのM行×N列のマトリクス5424からなり、この例では、M=3、N=4である。ここで、入力2進ストリングのビット数が X_L ビットとすると、Mは次式で求めることができる：

$$M = \log_2 (X_L + 1)$$

ただし計算結果は大きい側の最寄りの整数に丸める。また、Nは：

$$N = X_L$$

である。

マトリクス5424はXで表されている2進ストリングを入力してYで表されている2進数を出力する。この出力はその2進ストリング中の「1」の数を示す。Zで示されているもう1つの出力は最小化された「1」計数回路で用いられるが、これについては以下に説明する。

上記マトリクス5424の各カウントセル5426a～5426lはANDゲート及びXORゲート（排他的OR）を有する。例えば、カウントセル5426aはANDゲート5428a及びこれと結合されたXORゲート5430aを有する。ANDゲート5428aのようなANDゲートは、そのすべての入力が論理レベル

「1」である場合及びその場合に限って出力が論理レベル「1」となるAND機能を果たす。ANDゲート5428aは入力5432a及び5434aと出力5436aを有する。従って、入力5432a及び5434aの論理レベルがどちらも「1」の時出力5436aも「1」になる。また、入力のどちらかが「0」であれば出力5436aは「0」である。

XORゲートは、その入力に奇数の「1」が印加されている時のみ出力に論理レベル「1」を生じさせる。例えばXORゲート5430aはその入力5440a及び5442aのどちらか一方だけが「1」になっている時のみ出力5438aが「1」になる。

カウントセル5426aにおいてはマトリクス5424の他のすべてのカウントセル同様に、ANDゲート5428aはXORゲート5430aに結合されている。ANDゲート5428aの入力5432aはXORゲート5430aの入力5440aに接続されている。ANDゲート5428aの入力5434aはXORゲート5430aの入力5442aに接続されている。以上の接続関係により、ANDゲート5428aにはXORゲート5430aと同じ入力が供給される。

カウントセル5426a乃至5426lはマトリクス5424の行及び列をなすように配列されている。ここで、カウントセル5426a、5426b及び5426eの相互接続を例に取って、マトリクス5424全体の相互接続構成を説明する。第54図に

示すように、カウントセル5426aはカウントセル5426bの左側でカウントセル5426eの上側の位置に配置されている。また、カウントセル5426aはカウントセル5426bに接続され、カウントセル5426bのXORゲート5430の出力5438はカウントセル5426aの入力5432a及び5440aに接続されている。さらに、カウントセル5426aは5426eに接続され、カウントセル5426eのANDゲート5428aの出力5436は、カウントセル5426aのANDゲート5428aの入力5434a及びXORゲート5430aの入力5442aに接続されている。ここで説明した相互接続関係は、行間接続についてはカウントセル5426aと5426eとの関係を用い、また列間接続については、カウントセル5426aと5426bとの関係を用いることによりマトリクス全体に拡張して適用することができる。また、マトリクス5424は行と列を入れ換えて実施することも可能であり、マトリクスそのものはこの発明の要旨から逸脱することなく他のマトリクスと置換することができる。

ここで、マトリクス5424の構成をさらに詳しく説明するために、行及び列に言及する場合は下記の約束に従うものとする：行には行番号(M-1)を付すと共に、列には列番号(N-1)を付し、最下行を行0、右端の列を列0とする。第54図の例では、Mが3、Nが4である。さらに、(x,y)の位置にあるカウントセルを見ると、座標X及びYはそれぞれカウントセルの列番号及び行番号を示している。例えば、カウントセル5426aは(3,2)に位置している。

このようにして、マトリクス5424は行及び列をなすように配列され相互接続されたカウントセル5426a乃至5426lからなり、行0には2進ストリングXが供給され、行1には行0のANDゲート出力が供給され、行2には出力Zが生じる。列に関しては、列0には信号伝播を開始するために何らかのソースから「0」が供給され、列1には行0のXORゲート出力が供給され、列3には2進ストリング中の「1」の数を示す出力Yが生じる。列0に与えられる論理レベル「0」は入力をハードウェア的に接地することにより得ることも可能である。

ここで説明の便宜上2進ストリング1101($X_3=1$ 、 $X_2=1$ 、 $X_1=0$ 、 $X_0=1$)がマトリクス5424の行0に供給されるものと仮定する。すると、カウントセル5426lのANDゲート5428の出力が「0」となり、カウントセル5426lのXORゲート5430の出力が「1」になる。このカウントセル5426lのXORゲート5430の論理レベル「1」は行0に沿って伝達され、各カウントセルのXORゲートの出力は対応するX入力に「1」が入る毎にトグル動作する。従って、図示の如く上記2進ストリングが供給された場合、カウントセル5426kのXORゲート5430の出力は論理レベル「1」のままであり、カウントセル5426jのXORゲート5430の出力は「0」になり、カウントセル5426iのXORゲート5430の出力は「1」になる。その結果、行0の出力は「1」、即ち $Y_0=「1」$ となる。

行1でも、各XORゲートの出力は同様にトグル動作する。即ち、カウントセル5426hのXORゲートの出力は、カウントセル5426lのANDゲート5428より「0」が供給されており、「0」である。カウントセル5426gのXORゲート5430の出力は、このセルにはカウントセル5426hのXORゲート5430及びカウントセル5426hのANDゲート5428より共に「0」が供給されており、「0」のままである。そのため、カウントセル5426fのXORゲート5430の出力は、カウントセル5426gのXORゲート5430より「0」が、カウントセル5426jのANDゲート5428より「1」が入力されており、「1」となる。また、カウントセル5426eのXORゲート5430の出力は、カウントセル5426fのXORゲート5430より「1」が、カウントセル5426iのANDゲート5428より「0」が入力されており、「0」となる。その結果、行1の出力は「1」、即ち $Y_1=「1」$ となる。

行2では、カウントセル5426dのXORゲート5430の出力は、ハードウェア結線による「0」及びカウントセル5426hのANDゲート5428からの「0」が入力されており、「0」である。カウントセル5426cのXORゲート5430の出力は、カウントセル5426dのXORゲート5430及びカウントセル5426gのANDゲート5428より共に論理レベル「0」が供給されており、「0」のままである。その結果、カウントセル5426a及び5426bのXORゲート5430の出力は共に「0」となり、行2の出力は「0」、即ち $Y_2=「0」$ となる。そのため、図示例の入力2進ストリング $X=1101$ に対するマトリクスの出力2進数Yは $Y=011$ 、即ち10進数の3となる。実際、この2進ストリング入力 $X=1101$ 中の「1」の数は3になっている。

ここで、「1」計数回路5320が入力を受けてクロック信号の必要なしに出力を出す非同期回路であるということは明らかである。従って、マトリクス5424では、入力が入ると同時に出力が発生し、信号はマトリクスを通じて伝播する。マトリクスを通しての最長伝播時間は、信号がカウントセル5426l、5426h、5426d、5426c、5426b、及び5426aを含む最長経路を通して伝播するのに要する時間であろう。

第54図のマトリクス5424は矩形状マトリクスで、多数の同一構成のカウントセル5426よりなる。このような性格のため、「1」計数回路はコンパクトに作ることができ、半導体マスク製造のためのレイアウトも容易である。しかしながら、マトリクス5424は、カウントセルまたはゲートあるいはその両方の数をさらに少なくすることにより、なお一層最小規模化することもできる。

第55図には、4ビット2進ストリング入力用の最小規模構成の「1」計数回路マトリクス5544が示されている。このマトリクス5544は相互接続されたカウントセル5546a乃至5546eを具備している。最小規模構成のマトリクスの場合、行数M及び各行のカウントセル数Nはそれぞれ下記の式により与えられる：

$$M = \log_2 X_n$$

計算結果は大きい側の最寄りの整数値に丸める。

$$N = X_r - 2^r$$

ただし、 X_r は入力 2 進ストリング X のビット数であり、 r は 0 から $(M-1)$ の範囲の行番号である。第 55 図の例では、入力 2 進ストリング X のビット数 X_r は 4 である。従って、上記の式を用いて、行数 M は 2 となる。第 1 行について N を計算すると $r=0$ であるから、 $N=3$ と求まる。第 2 行については、 $r=1$ より $N=2$ となる。フル構成のマトリクス (第 54 図) での 12 個のカウントセルに比して、このように第 1 行に 3 つ、第 2 行に 2 つ、合計 5 つのカウントセルしか持たない最小規模構成のマトリクスによって、4 ビットの 2 進ストリング中の「1」の数を計算することができる。

カウントセル 5546a 乃至 5546e は、各々、第 54 図に示すフルマトリクス 5424 のカウントセルと同様、AND ゲート 5548 及びこれに結合された XOR ゲート 5550 よりなる。カウントセル 5546c 乃至 5546e の入力には 2 進入力ストリング X が供給される。また、セル 5546a 及び 5546c の出力には出力 2 進数 Y が生じる。

第 55 図の例では、カウントセル 5546c の AND ゲート 5548 及び XOR ゲート 5550 の入力 5552 に X_0 が供給され、 X_2 はカウントセル 5546d の AND ゲート 5548 及び XOR ゲート 5550 の入力 5554 に供給される。また、 X_1 はカウントセル 5546e の AND ゲート 5548 及び XOR ゲート 5550 の入力 5556 に供給され、 X_0 はカウントセル 5546 の AND ゲート 5548 及び XOR ゲート 5550 のもう一方の入力に供給される。

2 進数出力 Y の最上位ビット Y_2 はカウントセル 5546a の XOR ゲート 5548 の出力 5560 に発生する。また Y_1 はカウントセル 5546a の XOR ゲートの出力 5562 に生じる。最下位ビット Y_0 はカウントセル 5546c の XOR ゲート 5550 の出力 5564 に生じる。

この最小規模構成のマトリクス 5544 は矩形マトリクスではないから、カウントセル間の相互接続関係は前述の矩形マトリクスと異なり変更されている。特に、フルマトリクスと対比して (x, y) の位置にカウントセルが欠けている場合、その位置のすぐ「下」の行のカウントセルがそのすぐ左のカウントセル $(x+1, y)$ の XOR ゲートの入力に接続される。また、フルマトリクスに比べて 2 つ以上のカウントセルが欠けている場合、例えば、座標 (x, y) 及び $(x+1, y)$ の 2 つのカウントセルがない場合は、 $(x+1, y-1)$ の位置のカウントセルの AND ゲートの出力を $(x+2, y)$ のカウントセルの XOR ゲートの入力に接続しさえすればよい。第 55 図の実施例では、 $(0, 1)$ 及び $(1, 1)$ の位置のカウントセルが欠如しているから、 $(1, 0)$ の位置のカウントセル 5546e の AND ゲート 5548 の出力は $(2, 1)$ の所のカウントセル 5546b の AND ゲート 5548 及び XOR ゲート 5550 の入力に接続されている。また、フルマトリクスの実施例と比べて、 $(0, 0)$ の位置のカウントセルも欠けているため、 X_0 は $(1, 0)$ のカウントセル 5546e の AND ゲート 5548 及び XOR ゲート 5550 の入

力 5556 及び 5558 に直接接続されている。さらに、座標 $(3, 0)$ のカウントセルも欠如しているから、出力 Y_2 としては $(3, 1)$ の所のカウントセル 5546a の AND ゲート 5548 の出力 5560 が直接用いられる。

前出の入力 2 進ストリング $X=1101$ ($X_3=1$ 、 $X_2=1$ 、 $X_1=0$ 、 $X_0=1$) を入力として用いると、カウントセル 5546e の AND ゲート 5538 の出力は「0」であり、同カウントセル 5546e の XOR ゲート 5550 の出力は「1」である。カウントセル 5546a の XOR ゲート 5550 の出力論理レベル「1」は行 0 を介して伝播し、各カウントセルの XOR ゲートの出力は各々対応する XN 入力「1」となる毎にトグル動作する。従ってカウントセル 5546d の XOR ゲートの出力は「0」となり、カウントセル 5546c の XOR ゲート 5550 の再度出力は「1」になる。その結果、行 0 の出力としては「1」が得られ、 $Y_0=1$ となる。

2 番目の行 (行 1) では、カウントセル 5546b の AND ゲート 5548 の出力 Z は、このセルにはカウントセル 5546e の AND ゲート 5548 より「0」が供給されており、「0」である。カウントセル 5546b の XOR ゲート 5550 は、カウントセル 5546e より「0」が、またカウントセル 5546d より「1」が供給されており、「1」を出力する。カウントセル 5546a の XOR ゲート 5550 は、カウントセル 5546c より「0」が、カウントセル 5546b より「1」が供給されており、「1」を出力する。その結果、行 1 の出力は「1」で、 $Y_1=1$ となる。さらに、カウントセル 5546a の AND ゲート 5548 の出力である Y_2 は「0」となる。従って、出力 2 進数 Y は $Y=011$ となり、入力 2 進ストリング $X=1101$ 中に 3 つの「1」があることを示す。

上記マトリクス 5544 は、破線で描かれているセル 5546b の AND ゲート 5548 のような一部の論理ゲートを省くことによりさらに小規模化することが可能である。この AND ゲート 5548 の出力 Z は出力 2 進数 Y を組み立てるのにはふようであるから、このゲートは省略することができる。従って、最小規模構成のマトリクスでは、同じ行中の欠如したカウントセルのすぐ隣りのカウントセルの AND ゲートは省略可能であり、これによって「1」計数回路のサイズを一層縮小することができる。

ここで、この発明の範囲が本願で開示説明する回路の実施例に限定されるものではないということ言うまでもない。特に、上記以外の実施例としては、当技術分野では良く知られているようにここで説明した実施例に基づきブールロジックにより誘導可能な回路の実施例等が含まれる。例えば、AND ゲート 5548 のような AND ゲートは NAND ゲートにインバータを結合しても全く同等に実施することが可能である。さらに、当技術分野では周知のド・モルガンの定理により AND 機能は OR ゲートの出力にインバータを結合すると共に、その OR ゲートの入力を反転させるやり方でも実施可能である。上記のようにこの発明より誘導可能な代替的回路もこの発明の範囲内に含まれるものである。

次に、第56図には、この発明の文字認識の応用例が示されている。ピクセルのマトリクス5666は文字「F」を形成するように配列された「0」及び「1」で構成されている。このマトリクス5666のピクセルは前述の画像処理デバイスで収集し、画像データメモリに記憶したものであってもよい。ピクセルマトリクス5666は、行方向及び列方向に処理され、それぞれ各行の「1」の数の計数値（行方向カウント）5668及び各列の「1」の数の計数値（列方向カウント）5670が得られる。行方向カウント5668はピクセルマトリクス5666の各行を2進ストリング入力Xとして「1」計数回路に供給することにより得られる。このように、各行毎に「1」の数の計数値が得られる。第56図の例の場合、キャピタル文字「F」の最初の2行には「1」のピクセルはない。行3にはこの文字の最初の横線を形成する4つの「1」がある。行4には「1」は1つしかない。また、行5には文字「F」の2番目の横線を形成する3つの「1」があり、行6及び7にはいずれも「1」が1つずつ含まれている。

同様にして、列方向カウント5670はピクセルマトリクス5666の各行を「1」計数回路に供給することにより得られる。列1及び2には、どちらも「1」に入っていない。列3には文字「F」の縦線を形成する5つの「1」があり、列4及び列5にはどちらも2つの「1」が入っている。また、列6には「1」は1つしかなく、列7及び8には「1」は全くない。

このようにして、この発明によれば、パターン認識システムでは、すべての文字、キャラクタ及びどのようなイメージであっても行方向カウント及び列方向カウントをヒストグラムとして作り、記憶することができるので、これらを新しい文字画像サンプルに対する比較基準として用いることが可能である。

この発明の上記実施例では、入力2進ストリングの「1」の数を計数するが、「1」計数回路マトリクスの入力にインバータを付加することにより2進ストリングの「0」の数を計数するよう動作する「0」計数回路を用いた実施例ももとより可能である。このような「0」計数回路はこの発明の他の実施例の1つであり、この発明の範囲内に含まれるものである。

以上、この発明について詳細に説明してきたが、特許請求の範囲に記載したこの発明の趣旨並びに範囲を逸脱することなく数多の変更、置換及び改変を行うことが可能なことは明らかである。

各種プロセッサの詳細説明

以下、第29図乃至第45図を参照しつつマスタープロセッサ、パラレルプロセッサ、及び転送プロセッサについて詳細に説明する。この説明は特許請求の範囲に記載した本願発明の動作、作用を理解する上においては必ずしも必要ではないが、個々の具体的な実施例を得る上においては有用であろうと思われる。実際に使用されるシステムの詳細はそのシステムの要件によって決まり、以下

に論ずる内容からも大きく異なってくる場合もあり得る。

パラレルプロセッサ マスタープロセッサ

第29図には、同期及び色々なパラレルプロセッサ間に流れるその他の情報の制御を含め、イメージシステムプロセッサの動作を制御するのに用いられるマスタープロセッサ12の詳細な構成が示されている。マスタープロセッサ12は、命令を実行するが、その命令はオブコード回路2911（オブコード:OPCODE;オペレーションコード）及びレジスタファイル2901によって制御されるオブコードを有する32ビットワードを用いることができる。プログラムカウンタ（PC）2903は制御ロジック2904の制御下において動作し、バス172からオブコードレジスタ2911への命令のローディングを制御する。制御ロジック2904は命令を解釈し、マスタープロセッサ12でのオペレーションを与えられた命令に基づき制御する。

整数演算装置（ALU）2902の他、このマスタープロセッサには2つの部分からなる浮動小数点演算装置が設けられている。これら2つの部分とは、1つはマルチプライヤ2905、正規化回路2906及び指数加算器（EXP）2907で構成された浮動小数点乗算器であり、他の1つは事前正規化回路（PRENORM）2908、演算装置（AU）2909及び事後正規化シフタ（SHIFT-NORM）2910よりなる浮動小数点加算器である。

プログラムカウンタ2903は32ビット命令を読み込む必要がある時、バス172に沿ってアドレス出力を供給するために用いられる。整数演算装置2902は、オブコードレジスタ2911により解釈された命令に従って動作して、マスタープロセッサの外部に接続されるキャッシュメモリからのデータ読み込みを制御するアドレスを発生することができ、そのアドレスはバス171を介して出力される。これに対するデータは、キャッシュメモリよりバス171のデータ部を通して供給され、レジスタファイル2901に記憶される。

上記の命令バス172及びデータバス171はそれぞれアドレス部及びデータ部よりなっている。命令バス172の場合、アドレス部はプログラムカウンタ2903からアドレスを受け取り、データ部はオブコードレジスタ2911にデータ（命令）を取り込むよう接続されている。また、データバス171では、アドレス部は整数演算装置2902の出力からアドレスを受け、データ部はレジスタファイル2901に接続されて、データは書き込みサイクルならばレジスタファイル2901から出、読み込みサイクルならばレジスタファイル2901へ入れられる。

マスタープロセッサ12の各構成要素間の相互動作の様子は当技術分野においては周知である。たとえば、グラフィックスプロセッサの動作形態の一例が「浮動小数点コプロセッサ（双対プロセッサ）」という名称のコール・ガッターグ、デビッド・ガリー、及びジェリー・ヴァ

ン・エイケンによる同時係属米国特許出願（出願番号第 387,472;1989年7月28日出願、本願中に引用）に開示されている。

パラレルプロセッサの動作

第1図及び第2図に示す4つのパラレルプロセッサ (PP) 100~103はシステムオペレーションの大半を実行する。これらのパラレルプロセッサは各々高度の並行処理能力を有して、各サイクル毎に限定命令セット計算機のようなオペレーションを汎用機に倣って実行することができると共に、膨大なデータ処理能力を有し、特に画像/図形処理においてそのデータ処理能力を発揮する。

これらの各パラレルプロセッサは、クロスバスイッチを介してメモリへのアクセスを1サイクル当たり命令について1回、データについて2回、合計3回行うことができる。また、各パラレルプロセッサは各サイクル毎にマルチプライ（情報）演算及びALU（演算・論理装置）オペレーションを実行することができると共に、次の2つのデータ転送のためのアドレスを生成することができる。そして、効率的なループロジックにより3つのネストされたループ（入子形ループ）のサイクルオーバーヘッドをゼロにすることができ、論理値「1」を扱うための特殊なロジックが組み込まれており、さらにALUはパックされたピクセルに対するオペレーションのためにスプリット（分割）可能である。

前にも述べたように、パラレルプロセッサは、使用上のフレキシビリティを確保するため、同じ命令によりオペレーションを実行する（SIMD;単一命令多重データモード）よう構成することもできれば、独立命令ストリームにより実行する（MIMD;多重命令多重データモード）よう構成することも可能である。MIMDモードでは、パラレルプロセッサを互いにロックステップ関係でランさせ、プロセッサ間で効率的な同期データ転送を行うことができる。

偶然同じメモリを同時アクセスしてしまうのではないかという気苦労からプログラマをかいほうするために、クロスバスイッチにはコンテンション優先順位指定ロジックが内蔵されており、またパラレルプロセッサにはリトライロジックが内蔵されている。

メモリアクセス100~103は同じ論理設計になっているが、システム内部での接続関係で2つの差異の特徴がある。まず、これの各パラレルプロセッサにはハードワイヤード方式により一意の2ビット識別番号が供給され、これによってプログラムはアドレスのような各パラレルプロセッサ固有の情報を生成することができる。次に、SIMD用として構成された場合、1つのパラレルプロセッサ、即ちPP100が「マスター」SIMDマシンとして動作し、すべてのパラレルプロセッサに代わって命令フェッチを実行する。それ以外のパラレルプロセッサは「スレーブ」マシンとして動作し、単に与えられた命令ストリームを実行するだけである。

内部インターフェース

第30図に示すように、パラレルプロセッサ100~103は、命令ポート3004、グローバルポート3005及びローカルポート3006等の汎用のインターフェース並びにプロセッサ間通信リンク40を介してシステムの他の各部に接続されている。

命令ポート3004はMIMDモードにおいてはそれ自体の命令RAM10-1（10-5,10-9または10-14）に接続され、SIMDモードにおいては、他のパラレルプロセッサの命令バスに接続される。SIMD用として構成された場合は、「マスター」SIMDパラレルプロセッサ100のみがその命令バス上にアドレスを出力する。また、命令ポート3004は転送プロセッサ11にキャッシュミス情報を伝送するのにも使用される。

グローバルポート3005は、クロスバスイッチの全長に亘って走るパラレルプロセッサ専用のバスに固着されている。このバスを経て、パラレルプロセッサはクロスバスイッチが接続されたメモリ10のどのRAMにもアクセスすることができる。データの転送サイズは、一般に8,16または32ビットである。このポートに関連するコンテンション（競合）検出信号3210はクロスバロジックによりドライブされ、リトライを実行しなければならない時そのことを指示する。

ローカルポート3006は機能的にはグローバルポート3005と類似しているが、これがアクセスできるのは、各パラレルプロセッサに物理的に対向する位置のクロスバスイッチが接続されたRAMだけである。しかしながら、SIMDモードにおいては、4本のローカルPPバス6を直列に接続して「共用」読み込みを指定し、すべてのパラレルプロセッサ（またはその部分集合）に同じデータを供給することができる（RAM10-0,10-2,10-3または10-4の中の1つから）。その場合は、「マスター」SIMDパラレルプロセッサ100のみがデータのアドレスを供給する。

MIMD構成においては、PP（パラレルプロセッサ）プログラムをロックステップ関係で実行できるようになっている。それには、プログラマがコードの該当部分にLCK及びVULCKのパウンドを付すことにより指示する。各パラレルプロセッサにつき1出力ずつからなる4つの信号3020はこれらのパラレルプロセッサ間に伝達されて、各パラレルプロセッサがこのようなコード部分に遭遇している時そのことを指示する。パラレルプロセッサはこれらの信号をテストすることによってコードを同期して実行することができる。

上に述べたように、グローバルポート3005及びローカルポート3006には、コンテンションが起こった時及びリトライが必要な時を知らせるための信号3210及び3211（第32図）が供給される。SIMDモードの構成になっている場合は、コンテンションの問題がすべて解消されるまですべてのパラレルプロセッサは命令実行を休止しな

ればならない。そのために、すべてのパラレルプロセッサ間には、いずれかのパラレルプロセッサがコンテンションを検出した時アクティブ化される信号3007が伝送されるようになっている。次の命令は、この信号がイナクティブ化された時のみパラレルプロセッサによってロードされる。この信号は、「マスター」SIMDパラレルプロセッサ100がキャッシュミスを検出した時にもアクティブ化される。MIMDモード構成では、信号3007は無視される。

SIMD構成の場合は、パラレルプロセッサ間にスタックコヒーレンシー（コンシステンシー）が維持されなければならない。従って、条件付きコールを行う時には、「マスター」SIMDパラレルプロセッサ100から「スレーブ」SIMDパラレルプロセッサ101～103へその条件が真であって、「スレーブ」パラレルプロセッサ101乃至103は戻りアドレスをプッシュすべきであるということを指示する信号3008が必要である。

これ以外にSIMDコヒーレンシーが維持しなければならないのは、割込みが起こった時である。この条件を取り込むために、「マスター」SIMDパラレルプロセッサ100によってアクティブ化される信号3009があり、「スレーブ」SIMDパラレルプロセッサ101～103はこの信号を常時監視する。パラレルプロセッサ100～103はすべてこの信号がアクティブ状態の時割込み疑似命令シーケンスを実行する。

もう一つのSIMD割込み関連信号3010は、「マスター」パラレルプロセッサ100に「スレーブ」パラレルプロセッサ101～103がイネーブル化された割込みを保留しているということを示す。「スレーブ」パラレルプロセッサ101～103は、常時割込まれることを予期すべきではないため、この信号によってSIMDタスクに何らかの不具合が起こったことを指示することができる。

各パラレルプロセッサには沢山の割込み信号3011が供給される。これらの割込み信号は1つのパラレルプロセッサがメッセージ伝達のために他のパラレルプロセッサにより割り込まれるのを可能にするためのものである。マスタープロセッサ12もメッセージ伝達のため同様にパラレルプロセッサに割り込むことができる。また、マスタープロセッサ12は、新しいタスクを与えるためにも各パラレルプロセッサに割込みを掛けることができる。SIMDの場合、「スレーブ」パラレルプロセッサ101～103中の割込みロジックはスタックコンシステンシーのためにアクティブ状態に保たなければならない、割込みは若干異なる方法で処理されるが、これについては後述する。

パラレルプロセッサはパケット要求が必要な時、信号3012によって転送プロセッサに知らせる。転送プロセッサはパケット要求が処理された時もう一つの信号3013によってそのことを指示する。SIMD構成の場合、「マスター」パラレルプロセッサ100だけが転送プロセッサに対してパケット要求を出力する。

内部構成

パラレルプロセッサのバス構成が第30図に示されている。パラレルプロセッサは3つの主要装置、即ちプログラムフローコントロールユニット（プログラムフロー制御装置）3002、アドレスユニット3001及びデータユニット3000で構成されている。以下、これらの各装置について個別に説明する。

プログラムフローコントロール（PFC）ユニット3002は、第31図に示すように、プログラムカウンタ3100に関連するロジック、即ち命令キャッシュ制御ロジック3101、ループ制御ロジック3102、ブランチ/コールロジック3103及びPP同期ロジック3104を有する。このロジックは、パラレルプロセッサの命令RAM10-1, 10-5, 10-9または10-14からのオペコードのフェッチングを制御する。また、キャッシュミスが起こると、このロジックはコードをフェッチすることができるようセグメントアドレス及びサブセグメント番号を転送プロセッサに伝送する。

PFCユニット3002内には命令パイプライン3105がある。従って、PFCユニット3002はアドレスユニット3001及びデータユニット3000を制御するのに必要な信号3112を発生する。あるオペコードにより指定された即値データはやはり命令パイプラインから抽出され、必要に応じてデータユニットに送られる。

PFCユニット3002には、割込みイネーブル（INTEN）3107、割込みフラグ（INTFLG）3106及び割込みベクトルアドレス生成ロジックも設けられている。このロジックはベクトルを読み込み、プログラムカウンタ3100及びステータスレジスタ（SR）3108の内容をセーブし（退避させ）、また割込みルーティンへ分岐するために、アクティブな割込みを優先させ、疑似命令のシーケンスをパイプライン3105に注入する。

パケット要求ハンドシェイク信号3102及び3103もPFCユニット3002に接続される。

PFCユニット3002はパラレルプロセッサの一部であり、その動作態様はSIMDモードの場合各パラレルプロセッサ間で異なる。「マスター」SIMDパラレルプロセッサ100は多少なりとも通常の如く動作するが、「スレーブ」パラレルプロセッサ101～103は各々のキャッシュロジック3018をディスエーブル化し、現在フラグ3109をフラッシュする。ループロジック3102、同期ロジック3104及びパケット要求信号3102, 3103もディスエーブル化される。割込みロジックの動作態様は、すべてのパラレルプロセッサの動作態様が同等になるよう修正される。

第32図に示すアドレスユニット3001は同一構成の2つのサブユニット3200及び3201を有し、これらの各サブユニットは、クロスバスイッチが接続されたRAM10におけるデータ記憶場所の16ビット・バイトアドレスを生成することができる。各サブユニット内には、4つのアドレスレジスタ3202、4つのインデックスレジスタ3203、4

つの修飾レジスタ3204、モジュロレジスタ3205及びALU3206が設けられている。オペコードで3つの並列データアクセスが指定されると、サブユニット3200はグローバルポート3005を介してアドレスを出力し、もう一方のサブユニット3201はローカルポート3006を介してアドレスを出力する。アクセスが1つだけ指定された時は、単一の共用SIMD読み込みが指定されていない限り、サブユニット3200、3201のどちらからアドレスが出力されてもよい。単一の共用SIMD読み込みが指定されている場合は、「ローカル」サブユニット3201よりアドレスを供給する必要がある。

また、アドレスユニット3001は、グローバルバス3005、ローカルバス3006のどちらかまたは両方にコンテンションが検出されるとリトライをサポートする機能をも有する。

アドレス指定モードには、アドレスレジスタ修飾の有無に従い、また短期即値によるかインデックスレジスタによるかでプレインデキシング (PRE) 及びポストインデキシング (POST) がある。アドレスは、さらに、2の累乗モジュロの有無、ビット反転アドレス指定の有無、及び共用SIMD読み込みによって、データ空間またはI/O空間に入るよう修飾することができる。

アドレスユニット3001はまたグローバルポート3005またはローカルポート3006上のアライナ/抽出器 (ALIGN/EXTRACT) 3003 (第30図) をも制御する。これらのアライナ/抽出器3003は基本的にはRAM10へまたはRAM10からのバイト、ハーフワードあるいはワードの転送を行わせるバイトマルチプレクサである。また、これらのアライナ/抽出器3003は非整合 (ただしバイト整合された) ハーフワードまたはワードをロードあるいは記憶させる。必要ならば、ロードのサインエキステンション (符号拡張) も可能である。

データユニット3000 (第33図) は、8つのマルチポートデータレジスタ3300、フル32ビットのパレルシフタ (B.S.) 3301、32ビットALU (演算・論理装置) 3302、左右両端「1」/「1」の数ロジック3303、除算反復ロジック及び16×16シングルサイクル乗算器 (MULT) 3304で構成されている。また、データ伝送のため数個のマルチプレクサ3305～3309が具備されている。

さらに、多重ピクセル演算が可能のように特殊な命令が入れられている。ALU3302は2つまたは4つの同等の部分に分割可能で、これらの部分により加算、減算、比較を実行することができる。これらの演算に続いて、飽和、最大/最小、及び透過性を実行させるマージオペレーションを行うことができる。これと同じロジックを用いると、色膨張、色圧縮及びマスキング等のオペレーションも容易化することが可能である。

データユニットの命令はすべてシングルサイクルで実行され、またすべてレジスタ・ツー・レジスタオペレーションである。これらの命令はすべて、クロスバスイッ

チに接続されたメモリ10からまたはメモリ10へ、1つまたは2つの別個にコード化されたロードまたは記憶 (ストア) をデータユニットのオペレーションと並行に行うことを可能にする。即値命令が指定されると、それはオペコード中の並列移動を置換する。これら種々のオペレーションは上記8つのデータレジスタ3300以外のレジスタを用いても実行することができるが、その場合即値命令に関しては並列移動を指定することはできない。

バス構成

第30図に示すように、パラレルプロセッサにはそのデータバスの全長に亘って走る4本のバス3014～3017が設けられている。これらのバスはすべてデータの移動に用いられ、バス数 (及びレジスタの読み出しポート、書き込みポートの数) とデータユニットのオペレーションに許容されるソース及び行き先とのバランスを考慮して適切な数のバスが設けられる。

左端のバス3014は16ビットの即値データ (左/右の位置調整及び符号拡張後) データユニット3000へ伝送する。また、このバスは即値データをALU3302を通過させ、そこからレジスタ書き込みバス3016へロードするのにも用いられる。

左から2番目のバス3015はアドレスユニット3001あるいはPFCユニット3002のいずれかのレジスタソースからデータユニット3000へデータを伝送する。また、このバスはグローバルポート3005を介してメモリ10へ送られるストアのソースデータを運ぶのにも使用され、さらにALUオペレーションと並行して起こるレジスタ・ツー・レジスタ移動のソースをも運ぶ。

その右隣のバス3016は、メモリ10からグローバルポート3005を介していずれかのレジスタへロードするため、及びデータユニットのオペレーションの結果をいずれかのレジスタへ書き込むのに使用される。このバスは、パイプラインがコンテンション、同期またはキャッシュミスの間休止する際ロードデータを一時的に保持するラッチ3018を具備している。

右端のバス3017は、メモリ10からまたはメモリ10へデータユニットのレジスタ3300のロードまたはストアを行うために専らローカルポート3006によって使用される。このバスはデータユニットのレジスタ3300以外のレジスタは全くアクセスすることができない。また、このバスは、パイプラインがコンテンション、同期又はキャッシュミスの間休止する際ロードデータを一時的に保持するラッチ3019を具備している。

パイプラインの概要

パラレルプロセッサのパイプラインはフェッチ、アドレス及び実行と呼ばれる3つのステージを有する。以下、これらの各パイプラインステージについてそれぞれ簡単に説明する。

「フェッチ」：プログラムカウンタ3100に入っているアドレスがセグメントレジスタ3110の内容及び現在のフ

ラグ3109と比較され、命令があればフェッチされる。プログラムカウンタ3100は事後インクリメントされるかまたはループスタートアドレス (LSA) 3111よりロードされる。MIMD同期がアクティブ状態であれば、命令フェッチの許可または禁止が行われる。

「アドレス」：命令が1つまたは2つのメモリアクセスを要求すると、アドレスユニット3001はこのステージの間に所要アドレスを生成する。そのアドレスの上位の5ビットはコンテンション検出/優先順位指定のためクロスバスイッチ20に供給される。

「実行」：レジスタ・ツー・レジスタのデータユニット3000のオペレーション及びその他のデータ移動はすべてこのステージ中に行われる。クロスバスイッチアドレスの上記の残りの11ビットはRAM10へ出力され、データ転送が実行される。コンテンションが検出されると、それが解消されるまでこのステージがリピートされる。プログラムカウンタ3100が行き先（即ちブランチ、コールまたはリターン）として指定されると、PC3100はこのステージ中に書き込まれ、その結果2つの命令の遅延スロットを生成する。

メモリ

各パラレルプロセッサは下記の3つの別個のメモリ空間をアクセスする：

◇64メガバイトのオフチップのワード整合メモリ空間（オンチップキャッシュより）；

◇64キロバイトのオンチップのクロスバスイッチに接続されたメモリ10、この空間はデータ空間と称する；

◇64キロバイトのオンチップI/O空間、この中にはパラメータRAM、メッセージレジスタ及びセマフォフラグがある。

各パラレルプロセッサ100～103のI/O空間は互いに分離されているので、コードはI/O空間をアクセスする時各パラレルプロセッサに対して一意のアドレスを計算する必要はない。従って、各パラレルプロセッサは同じ論理アドレスの自己のパラメータRAMを見ることになる。これはメッセージレジスタ及びセマフォフラグについても同じである。しかしながら、マスタープロセッサは各パラレルプロセッサのI/O空間を一意にアドレスすることができる。

上記メモリの64キロバイトはあくまでも1つの実施例であって、これを拡張したり変更したりすることが可能なことはもちろんである。

プログラムフローコントロールユニット

プログラムフローコントロール (PFC) ユニット3002 (第31図) はほとんどフェッチパイプラインステージで動作し、命令パイプラインのローディングに影響を及ぼす。しかしながら、命令パイプラインはPFCユニット3002内にあるので、このユニットはアドレス及び実行パイプラインステージの時もアドレスユニット3001及びデータユニット3000に対して信号3112を発しなければなら

い。また、PFCユニット3002はアドレスユニット3001からコンテンションが起こったということを示す信号を受け取り、パイプラインを休止する。

キャッシュ制御

512-命令キャッシュは4つのセグメントを有し、これらの各セグメントは各々4つのサブセグメントよりなる。従って、各サブセグメントは32の命令を含むことになる。各サブセグメントには1つの現在フラグがある。プログラムカウンタ3100は24ビットであるから、セグメントレジスタ3110は各々17ビットである。また、命令オペコードは32ビット幅である。

命令RAMはアクセスするのに用いられる9ビットワードアドレスはプログラム3100の下位の7ビットとセグメントアドレス比較ロジック3113からの2ビットから作られる。この比較ロジック3113はRAMアクセスを著しく遅延させることがないように迅速に動作しなければならない。

プログラムカウンタ3100の上位17ビットがセグメントアドレスレジスタ3110の1つと一致しない場合は、セグメントミスが起こっている。最低使用頻度のセグメントが選ばれ、ロジック3114によりトラッシュされ、そのサブセグメントの現在フラグ3109がクリアされる。しかしながら、プログラムカウンタ3100の上位17ビットがセグメントアドレスレジスタ3110の1つと一致し、それに対応するサブセグメントのフラグがセットされていない場合は、サブセグメントミスが起こっている。

いずれかの型のキャッシュミスが起こるとパイプラインは休止され、キャッシュミス信号3115が転送プロセッサ11へ送られる。キャッシュミスアクノリッジ信号が転送プロセッサ11より供給されると、プログラムカウンタ3100の上位17ビット及び充填されるべきサブセグメントを表す4ビットが転送プロセッサバスに出力される。

(これには、パラレルプロセッサ命令バス (水平バス7)、及び転送プロセッサバス (水平バス0) の間のクロスバスイッチ接続0-3, 0-8, 0-13または0-18が必要である)。そして、転送プロセッサアクノリッジ信号3115はイナクティブ化される。そのサブセグメントが転送プロセッサ11によって充填されると、キャッシュフィルド信号3115がパラレルプロセッサに送られて、対応するサブセグメントの現在フラグ3109をセットし、パラレルプロセッサのキャッシュミス信号3115をイナクティブ化して、命令実行が再開される。

パラレルプロセッサがキャッシュミス要求の処理のために待機している時そのパラレルプロセッサに割込まれると、そのキャッシュミス処理は中止され、これによって望まれていないコードの不必要なフェッチを防止するようになっている。

SIMD構成においては、「スレーブ」パラレルプロセッサ101～103の現在フラグ3109がクリアされた状態に保たれ、キャッシュロジック3101は無視される。「スレー

10

20

30

40

50

ブ」パラレルプロセッサ101~103は、SIMD休止信号3007がイナクティブの時は常に命令（「マスター」パラレルプロセッサ100により供給される）をパイプラインにロードする。「マスター」パラレルプロセッサのキャッシュ3101は通常の如く動作するが、これもSIMD休止信号3007がアクティブの時は常にパイプラインを休止する。

（このような条件は「スレーブ」パラレルプロセッサ101~103がコンテンションを検出すると生じる）。MIMD構成においては、SIMD休止信号はすべてのプロセッサにより無視される。

パラレルプロセッサのキャッシュ3101をフラッシュする能力はマスタープロセッサ12によりアクセス可能なメモリマップドレジスタによって得ることができる。このような機能は選択されたパラレルプロセッサのすべての現在フラグをクリアする。

ループ制御

ここでは、専らアドレスユニット3001用のロジックを用いる代わりに、コンボリューションのようなオペレーションを速度を犠牲にすることなく適宜のアドレスによりコード化することができるようサイクルオーバーヘッド等で実行する3つのネスト化されたループが設けられている。

この特徴をサポートするために、16ビットの3つのループエンド（LE）値レジスタ3116~3118、16ビットの3つのループカウンタ（LC）レジスタ3119~3121、16ビットの3つのループリロード（LR）値レジスタ3122~3124、及び24ビットの1つのループスタートアドレス（LSA）値レジスタ3111等多数のレジスタが具備されている。これらの3つのループ通のスタートアドレスが共通になっているてんは制約要因であるが、この制約はもう2つ24ビットのループスタートアドレスレジスタを付加するだけで解消することができる。

ループレジスタ3111及び3116~3124をロードするのに必要な命令の数は、ループカウンタレジスタ3119~3121を、これに対応するループリロードレジスタ3122~3124が書き込まれる時常に、同時にロードすることによって減らすことができる。このやり方で最大3つの命令を節約することができる。この場合、セーブされたレジスタを復旧する、即ち文脈スイッチ後に復旧する時は、ループリロードレジスタ3122~3124をループカウンタレジスタ3119~3121よりも前に復旧しなければならない。

ステータスレジスタ3108（第34図）には、ループを幾つアクティブ化する必要があるかを示す2ビット（ビット25, 24）が用意されている。（最大ルーピング深さビットMLD）。また別の2ビット（ビット23, 22）が現在のルーピング深さを示すために用意され（現在ルーピング深さビットCLD）、2ビットディクレメンタの形で実施されている。これらのビットはループエンドアドレスレジスタ3116~3118の中のどれをプログラムカウンタ3100と比較すべきかを指示する。これらのCLDビットはリ

セットにより、またステータスレジスタ3108がプッシュされていれば割込みによってゼロにクリアされる（どのループもアクティブ状態でなくなる）。ループには最も外側のループを1として1から3までの番号が付されている。ユーザはループロジックをアクティブ化するためにはMLDビット及びCLDビットを所望の値にセットしなければならない。すべてのループが完了するとCLDビットは0になる。

CLDビットはフェッチパイプラインステージの間ループロジックにより自動的にディクレメントされるから、ループ内の最後の2命令の間はステータスレジスタ3108に書き込みを行うべきではない。

ループロジック3102がいったんアクティブ化されると（CLDビットにゼロ以外の値があることにより）、そのCLDビットにより指示される16ビットのループエンドアドレスレジスタ（3116~3118の中の1つ）が各命令フェッチの間インクレメントされていないプログラムカウンタ3100と比較される。この比較で一致の結果が得られる且つ対応するループカウンタ（3119~3121の中の1つ）が1でなければ、ループスタートアドレスレジスタ3111の内容がプログラム3100にロードされ、ループカウンタ（3119~3121の中の1つ）がディクレメントされ、MLDビットがCLDビットにコピーされる。

しかしながら、インクレメントされていないプログラムカウンタ3100とループエンドアドレスレジスタ（3116~3118）は内容が等しく、対応するループカウンタ（3119~3121）の内容が1であると、CLDビットが1だけディクレメントされ、対応するループカウンタ（3119~3121）がそれに対応するループリロードレジスタ（3122~3124の1つ）よりリロードされ、プログラム3100は次の命令へインクレメントする。

ループエンドアドレスレジスタ3116~3118は16ビットしかないから、ループは64K命令より長くはならないということになる。また、現在使用中のループエンドアドレスレジスタ（3116~3118の中の1つ）の16ビット値の形でループ外へのブランチまたはコールが偶然現れないかどうか注意すべきである。ユーザはCLDビットにゼロをセットしてこの点につき何ら問題がないことを確認するべきである。ループエンドアドレス比較は、ブランチまたはコールの2つの遅延スロット命令の間割込みからのリターンを助けるためにディスエーブル化される。

フェッチパイプラインステージの間はループロジックが作動するから、必要ならばルーピングとMIMD同期化を結合することができる。ループの割込みは上記同様問題ではない。SIMDにおけるルーピングは「マスター」SIMDパラレルプロセッサ100のループロジックによって制御される。この場合、「スレーブ」パラレルプロセッサ101~103のループロジックは、それらのプログラムカウンタ3100は無視されるのでやはり動作可能である。

上記の構成については種々の構成要素を置換した実施例が可能である。ややユーザに好意的なやり方は、3つの比較器付24ビットループエンドレジスタ及び3つの24ビットループスタートアドレスレジスタを具備することである。各ループはステータスレジスタ中の1つのビットによりイネーブル化することができる。

共通タスクに対して作用するMIMDプログラムを実行する時は、通常プロセッサ間で通信する必要がある。このシステムは、「ルース（疎）」通信のためのメッセージ伝送及びセマフォをサポートするが、密接に結合された（密結合）プログラムを実行する時は情報交換をサイクル単位ベースで行う必要がある。この場合は同期実行が役に立つ。

各パラレルプロセッサの同期ロジックSYNC/PP #3104のレジスタには各パラレルプロセッサに対して1つずつ4ビットが設けられている。ある1つのパラレルプロセッサを同期させる他のパラレルプロセッサは、そのプロセッサのレジスタのそれらのパラレルプロセッサに対応するビットに「1」を書き込んで指示する。同期させる予定の他のプロセッサについても、これと同様に各々のSYNC/PP #3104のレジスタの該当するビットをセットする。

同期して実行することが望ましいコードはLCK（ロック）及びULCK（アンロック）命令でバウンドすることにより指示される。LCKに続くULCKまでの命令（ULCKを含む）は他のパラレルプロセッサとロックステップ関係で実行される。従って、同期した各パラレルプロセッサでは、LCK命令とULCK命令との間の命令は同数でなければならない。

同期されたコードが実行されていると言う情報はステータスレジスタ3108の「S」ビット（ビット26）により記録される。このビットは、それぞれLCKまたはULCK命令のアドレスパイプラインステージのマスターフェーズまではセットまたはリセットされないが、LCKまたはULCK命令はスレーブフェーズの間次の命令のフェッチに影響を及ぼす。このビット26は、リセットにより、またステータスレジスタ3108がプッシュされている場合は割込みによってクリアされる。

パラレルプロセッサはLCK命令（アドレスパイプラインステージのスレーブフェーズによりデコードされる）に遭遇すると、同期されたコードを実行中であるということを示す信号40を他のパラレルプロセッサ100～103に出力する。次に、パラレルプロセッサは同期関係を欲する他のパラレルプロセッサからの入力同期信号をAND処理し、その結果これらのプロセッサがすべて同期信号40を出力している時のみ次の命令がパイプラインへフェッチされる。この動作は、同期しているすべてのパラレルプロセッサは同じ一致同期信号群がアクティブ状態でなければ処理動作を行わないので、これらすべてのプロセッサで同時に実行される。そのため、異なる2つの

MIMDタスクは、各々他方の同期信号を無視するので、同時にランすることができる。

また、同期されるのは命令フェッチであるから、同期されたコードに割込みを掛けることが可能である。この割込みはパラレルプロセッサの同期信号40を直ちにイナクティブ化させる。キャッシュミス及びコンテンションはほぼ類似の結果をもたらし、ハードウェアをインステップ状態に保持する。ただし、コンテンションの場合は、コンテンションが生じた命令に続く2つの命令がパイプライン中にフェッチされてからパイプラインが休止する。

アイドル（IDLE）命令を同期されたコードに入れて、あるパラレルプロセッサが割込みされた後その割込みルーティンから復帰するまで、同期されたすべてのパラレルプロセッサのオペレーションを休止させることも可能である。

同期されたコードに割込みを掛けることができるようにするため、いずれか1つのプロセッサのプログラムカウンタPC3100を行先として指定した命令は、すべて状態レジスタのSビットの作用を直ちにディセーブル化（無効化）するが（ULCK命令と同じタイミングで）、Sビットはセット状態を保持する。いったん2つの遅延スロット命令が完了すると、Sビット（26）の作用が再びイネーブル化（有効化）される。このメカニズムによれば同期された遅延スロット命令に割込みが掛けられなくても何ら問題はない。そのために、同期ロジック3104はブランチ（分岐）、コール（呼出し）及びリターン（復帰）を単一の命令として取り扱う（2つの遅延スロット命令が続くPC3100ロードとして実行される）。同期信号40は2つの遅延スロット命令の間イナクティブ化され、これらの命令は同期信号40を見ることなくフェッチされる。LCK命令が遅延スロットに入れられた場合、その効果は遅延スロット命令が実行された後に発揮される。同期されたループは、その「ブランチ」は実行ステージではなくフェッチパイプラインステージで働くので、通常のコード同様の挙動を示す。

同期の作用の仕方の一例が第23図に示されている。この場合、パラレルプロセッサ102（PP2）とパラレルプロセッサ101（PP1）は、A0とA1がそれぞれ各パラレルプロセッサ101及び102に同じアドレスを有していると仮定すると、各々のデータD0レジスタの内容を交換する。ここで、また、A0とA1はコンテンションを避けるために互いに異なるRAMを指示するものと仮定する。（同じRAMを指示しても有効に作動するが、余分のサイクルが必要である）。

この例では、PP1は、PP2がそのLCK命令に達してから1サイクル後にLCK命令に達する。このように、PP2は1サイクル待機する。そして、これらのパラレルプロセッサは同時に記憶を行うが、PP2は、ロード命令をフェッチする時キャッシュミスを起こす。両パラレルプロセッサ

サはこのキャッシュミスが転送プロセッサ11によって処理されるまで待機する。その後、これらのプロセッサは同時に且つ同じようにULCKをロードする。次いでPP1が命令4をフェッチする時にキャッシュミスを起こすが、これらのプロセッサは今度はアンロックされているので、PP2は妨害されることなく動作し続ける。

この簡単な例は命令1と2を組合せ、また命令2と3を組み合わせてることによって更に簡単化することができる。(即ち、LCK11 STの後にULCK11 LDWを続ける)。このやり方でまさしくロードは同期されるが、ここで必要なのはそのことだけである。

SIMDモードでの同期は、その方式自体がもともと同期性を含んでおり、LCK及びULCK命令が目的を持たず、従ってコード化されたとしても何ら作用しない。ステータスレジスタ3108のSビット(26)は、たとえプログラムで「1」にセットされているとしても何の効果もない。割込み及びリターン

プログラムカウンタ3100がロードされた後においては、2つの遅延スロットの間割込みはロックアウトされなければならない。これによってプログラムカウンタ3100の現在の値とブランチアドレスを共にセーブし、リターンで回復させなければならないというような事態を防ぐことができる。プログラムカウンタ3100のロードは遅延スロット命令の間禁止されるが、ユーザがそれでもこのようなロードを実行するような場合、ロックアウト期間は延長されない。そうでないと、割込みがいつまでもロックアウトされることが起こり得るからである。

多くのプロセッサとステータスレジスタ3108にはグローバル割込みイネーブルビット(27)(I)がある。ユーザは、このビットをセット/リセットすることによってマスタータスク割込み及びILLOP(違法オペコード)割込みを除くすべての割込みをイネーブル/ディスエーブル化することができる。このビット(27)は、リセットにより、またステータスレジスタ3108がブッシュされていれば割込み疑似命令によってクリアされる。

割込みからのリターンはシーケンスPOP SR, POP PC, DELAY1, DELAY2によって実行される。この場合、ステータスレジスタ3108のI(27), S(26), 及びCLD(23),

(22)の各ビットはDELAY2命令の前にPOP SRによってロードされるが、その作用はブランチ(POP PC)命令が完了するまで抑止される。これによってリターン完了前に作用が現れるのを防止する。

パラレルプロセッサ100~103は、各々最大16の割込みソースを用いることができるようになっている。これらの中8つのソースは割込みが指定されており、他のソースは将来の拡張のために残してある。ここで指定された割込みは下記の通りである。

マスタータスク マスタープロセッサがパラレルプロセッサ100~103に新しいタスクをランするよう要求する(常にイネーブル化(許可)される)

Illop 違法オペコードが検出された(常にイネーブル化される)

MIMDエラー 「マスター」SIMD PP100にのみ適用れる。3つの「スレーブ」PP101~103のすべてのイネーブル化された割込みのOR(論理和)

Illadd 存在しないオンチップアドレスがアクセスされた

コンテンション コンテンションが検出された。コンテンションを通常のやり方で解消した後割込みを行う

10 バケット要求 転送プロセッサがPPのバケット要求連係リストを使い切った(イグゾースト)

マスターメッセージ マスタープロセッサ12がPPのメッセージレジスタに書き込みを行う時生じる。

PP0 メッセージ PP0がPPのメッセージレジスタに書き込みを行う時生じる

PP1 メッセージ PP1がPPのメッセージレジスタに書き込みを行う時生じる

PP2 メッセージ PP2がPPのメッセージレジスタに書き込みを行う時生じる

20 PP3 メッセージ PP3がPPのメッセージレジスタに書き込みを行う時生じる

割込みレジスタ

割込みは2つのレジスタ、即ち割込みフラグレジスタ3106(INTFLG)及び割込みイネーブルレジスタ3107(INTEN)によって制御される。

割込みイネーブルレジスタ3107は、各々対応するビットがハードワイヤードで「1」にセットされたマスタータスク及びILLOP割込みを除き、各割込み毎に個別のイネーブルビットを有する。このレジスタはリセットによりオールゼロにクリアされる(「1」にハードワイヤードされたビットを除く)。ビット15~ビット0は使用されない。

30 割込みフラグレジスタ3106は各割込みソース毎に個別のイネーブルビットを有する。このフラグはそれぞれ単一サイクルの間アクティブ化されるソース信号によりラッチされる。このレジスタはリセットによりオールゼロにクリアされる。ビット15~0は使用されない。保留のマークが付されているビットも「0」にハードワイヤード接続される。すべてのフラグは「1」を書き込むことによってクリアされる。「0」を書き込んでも何の効果もない。これによって、必要ならば、割込みを生成する代わりにフラグをソフトウェアによってポーリング及びクリアすることができるようになっている。割込みが行われると、対応するフラグがハードウェアによって自動的にクリアされる。フラグがソースによってセットされようとしており、これと同時にクリアされようとしている場合は、セットが優先する。

50 割込みフラグレジスタ3106は、ステータスレジスタ3108のR((リストアドレス)ビット(19)がセットされておれば、通常のデータレジスタと同様「0」及び

「1」を用いて書き込むことができる。これによってタスク状態復元ルーティンは割込み状態を復元する事ができる。

ステータスレジスタ3108のIビット(27)をセットさせることにより割込みがイネーブル化(許可)されると、それらの割込みには優先順位が指定される。セットされるフラグを有するイネーブル化された割込みはすべて優先され、すぐ次の機会に割込みが生成される。この場合、下記のような3つの疑似命令のシーケンスが生成される:

① 割込みベクトルのアドレスを生成し、プログラムカウンタ(PC)3100内へフェッチし(まずPCをRET 3103へコピーする)、割込みフラグレジスタ3106内のフラグを同時に再セットされる場合を除きクリアする;

② RET 3103をプッシュする;

③ ステータスレジスタ(SR)3108をプッシュし、SR3108のS(26),I(27)及びCLD(22),(23)ビットをクリアする。また、これらのビットに対応する動作、機能を実行ステージが完了するまでディスエーブル化する。

コンテンション解消は、上記シーケンスによりサポートされなければならないので、実行には3サイクル以上が必要な場合がある。同様に、割込みルーティンの最初の2つの命令のどちらかに対するキャッシュミスは、パイプラインを休止させる。

割込みベクトルはパラレルプロセッサ自体のパラメータRAM10よりフェッチされる。これらのベクトルは、パラレルプロセッサ100~103の各々に対して同じ論理アドレスに存在するから、これらの各パラレルプロセッサ100~103の割込みロジックは同じベクトルアドレスを生成する。

割込みルーティンの最初の2つの命令はどちらもLCK命令であってはならないと言うのはパイプライン処理のためである。同様の理由で、上記の第3の疑似命令の実行ステージが完了するまで割込みロジックは割込み3106、同期ロジック3104及びループロジック3102をディスエーブル化しなければならない。これによって割込みルーティンの最初の2つの命令のフェッチングの間これらのロジックの機能がアクティブ化されるのを防いでいる。

割込みは、SIMDモードの場合MIMDモードの場合とは幾分異なる扱いがなされる。SIMDモードにおいては、スタックコヒーレンシーを維持するために、「マスター」パラレルプロセッサ100からスレーブ」パラレルプロセッサ101~103へは割込みを受けているということを指示する信号が送られる。この信号は「スレーブ」パラレルプロセッサ101~103に各々の割込み疑似命令のシーケンスを実行させる。プログラムカウンタ3100はいずれにしても無視されるので、どの割込みベクトルをフェッチするかは全く問題ではない。

また、SIMD構成では、逆に「マスター」パラレルプロセッサ100へ「スレーブ」パラレルプロセッサ101~103がイネーブル化された割込み事象を検出したということを示す信号を送る必要がある。その事象は、例えばコンテンション、違法アドレスアクセスあるいはメッセージ割込み等である。これらはいずれもエラーとなることはほとんど確実であるから、それらの事象は「マスター」パラレルプロセッサ100上で唯一の割込みレベルにより処理される。「スレーブ」パラレルプロセッサ101~103から「マスター」パラレルプロセッサ100へ1つの信号3010が供給されているが、これは全部の「スレーブ」パラレルプロセッサ101~103のイネーブル化割込みの論理和(OR)信号である。割込みを出しているスレーブ101~103は、「マスター」・ツリー・「スレーブ」への割込み信号3009が有効(アクティブ)になるまで割込み疑似命令を実行しない。

SIMD休止信号3007がアクティブの時割込みが発生すると(「マスター」パラレルプロセッサ100より)、「マスター」・ツリー・「スレーブ」割込み信号3009の送出は、休止の原因が取り除かれるまで遅延される。休止の原因がキャッシュミスであるば、そのキャッシュミスが解消されるとすぐに割込みを行うことができる。

ブランチ及びコール

ブランチ及びコールはプログラムカウンタ(PC)3100に書き込みを行うことにより実行される。PC3100は他のパラレルプロセッサのレジスタと同様にアドレス指定可能なレジスタであり、ブランチアドレスはPC3100に書き込まれ、PC+1という値がリターンアドレスレジスタ

(RET)3103にコピーされる。これはブランチが実際はコールであったばあいのリターンのために必要な値である。次に、このRETレジスタ3103は、コールの中に組み入れるために遅延スロット命令のいずれかによってスタック状にプッシュされるようプログラムされる。条件付きコールができるようにするために、条件付きでリターンアドレスをプッシュする命令が用意されている。これはブランチが行われている場合に実行されるだけである。

前にも述べたように、プログラムカウンタ3100を行き先として指定する命令は、2番目の遅延命令がフェッチされるまで割込みをロックアウトする。これによってブランチアドレスまたはリターンアドレスあるいはその両方が消失してしまうような問題を防止することができる。上記の期間中は同期もディスエーブル化されるが、これについても既に述べた。ループロジック3102が余りにも早くアクティブ化されることによる割込みからのリターンに付随する問題を防ぐため、上記2つの遅延スロット命令の間はループエンドアドレス比較もディスエーブル化される。

ステータスレジスタ

ステータスレジスタ3108はPFCユニット内に設けら

10

20

30

40

50

れ、構成は第34図に示されている。以下、各ビットの機能について説明する。

先ず、N（否定ビット）（31）は結果が否定の時一部の命令によりセットされる。ソフトウェアでのこのビットへの書き込みは通常の否定結果セッティング機構をオーバーライドする。

C（キャリービット）（30）はキャリーが起こった時一部の命令によりセットされる。ソフトウェアでのこのビットへの書き込みは通常の結果キャリーセッティング機構をオーバーライドする。

V（オーバーフロービット）（29）はオーバーフローが起こった時一部の命令により書き込まれる。これは永久ラッチされるオーバーフローではない。このビットの値は、これをセット/リセットする次の命令が実行されるまでしか保持されない。ソフトウェアでのこのビットへの書き込みは通常の結果オーバーフローセッティング機構をオーバーライドする。

Z（ゼロビット）（28）は結果がゼロの時一部の命令によりセットされる。ソフトウェアでのこのビットへの書き込みは通常のゼロ結果セッティング機構をオーバーライドする。

I（割込みイネーブルビット）（27）は、グローバル割込みイネーブルビットで、リセット及び割込みによりゼロにセットされる。これは割込みイネーブルビットがセットされたすべての割込みを許可する。通常のパイプライン遅延のために、このビットの値の変更は実行ステージが完了するまで何の効果ももたらさない。

S（同期コード実行ビット）（26）は、同期MIMDコード実行が稼働中であることを示すビットで、リセット及び割込みによりゼロにセットされる。命令は、同期ロジックSYNC/PP# 3104のレジスタ中の同期（SYNC）ビットにより指示されるすべてのパラレルプロセッサがアクティブな同期信号40を出力している時にフェッチされるだけである。このビットの値はSIMD構成では無視される。

MLD（最大ルーピング深さビット）（24）及び（25）は、何種類のレベルのループロジックが稼働しているかを示すビットで、リセットによりゼロにセットされる。00はルーピングがないことを示し、01はループ1だけ、10はループ1と2、11は3つのループ全部がアクティブであることをそれぞれ示す。

CLD（現ルーピング深さビット）（22）及び（23）は、現在どのループエンドレジスタがプログラムカウンタと比較されているかを指示するビットで、リセットによりゼロにセットされる。00はルーピングがないことを示し、01はループエンド1、10はループエンド2、11はループエンド3をそれぞれ示す。これらのビットは、リセットによりゼロにセットされ、ステータスレジスタ3108がプッシュされていれば割込みによりゼロにセットされる。

R（復元レジスタビット）（19）は、タスクスイッチ

後ハードウェアの状態を復元する時に用いられるビットで、リセットによりゼロにセットされる。このビットは、「1」にセットされていると、通常のレジスタと同様に「1」及び「0」による割込みフラグレジスタ3106への書き込みを可能にし、またメッセージレジスタを新しいメッセージ割込みを生じさせることなく復元させる。このビットは、また、同様の理由でステータスレジスタ3108のQビット（17）への書き込みを許可する。従って、このRビット（19）はタスク復元ルーティンでのみ用いられる。

U（アップグレードパケット要求優先順位ビット）

（18）は、通常のバックグラウンド優先順位のパケット要求をフォアグラウンドに上げるのに用いられ、リセットによってゼロにセットされる。このビットの値は転送プロセッサ11へ伝送され、Qビットの値と共に転送要求の優先順位を決定するために用いられる。このビットはソフトウェアによりリセットされるまでセット状態に保持される。

Q（待機パケット要求ビット）（17）は、パラレルプロセッサがパケット要求の待機中であるということを示すビットで、リセットによりゼロにセットされる。このビットはステータスレジスタ3108のPビット（16）に「1」が書き込まれた1サイクル後にセットされる。このビット（18）の値は転送プロセッサ11に伝送され、Uビットの値と共に転送要求の優先順位を決定するのに用いられる。このビットは、パラレルプロセッサのパケット要求の連係リストがいったん使い切られると、転送プロセッサ11によりクリアされる。このビットが転送プロセッサ11がこれをクリアしようとするのと同時にソフトウェアが（Pビット（16）を介して）これをセットしようとしている場合は、セットが優先される。このビットへの直接書き込みは、ステータスレジスタ3108のRビット（19）が、「1」または「0」を書き込むことができる時、セットされていない限り何の作用ももたらさない。このビットは不必要なパケット要求をデキューするのに用いることができるが、割り込まれたタスクを復元するのにより多く必要である。

P（パケット要求ビット）（16）は、ステータスレジスタ3108のQビット（17）をセットするのに用いられるワンショットの短サイクルビットであり、リセットによってゼロにセットされる。このビットは転送プロセッサ11へパケット要求を発動させる。このP/Qビットメカニズムは、パケット要求が読み出しと書き込みの間に転送プロセッサ11によりクリアされている場合に、偶発的にパケット要求を発動させることなくステータスレジスタ3108上で読み出し—変更—書き込みオペレーションができるようにするためのものである。

ステータスレジスタ3108の使用されていないビットはすべて「0」として読まれる。将来とも装置互換性（デバイスコンパティビリティ）を維持するために、これら

10

20

30

40

50

未使用ビットには「0」のみを書き込むべきである。
同期インディケータ

4つの同期 (SYNC) ビットはMIMDパラレルプロセッサをどのパラレルプロセッサに同期させたいかを指示するために用いられ、リセットによってゼロにセットされる。LCK及びULCK命令によりバウンドされたコードを実行する時は、対応する同期ビットの「1」により指示されたすべてのプロセッサが同期信号40を出力しない限り、命令フェッチは処理されない。これらのビットの値はSIMD構成では無視される。

2つのPP#ビットはパラレルプロセッサ100~103の各々に一意的に割り当てられている。これらのビットは、ソフトウェアがどのパラレルプロセッサをランしているのかを判断することができるよう、従って正しい一意アドレスを計算できるようハードワイヤード方式になっている。これらのビットへ書き込みを行っても何の作用ももたらさない。

これらのビットのコーディングは、00--パラレルプロセッサ (PP) 0 100, 01--PP1 101, 10--PP2 102及び11--PP3 103という関係になっている。PP0 100は「マスター」SIMDである。これに対応するパラレルプロセッサのローカルクロスバススイッチRAMのスタートアドレスは:0000h--PP0 100, 2000h--PP1 101, 4000h--PP2 102及び6000h--PP3 103である。

パイプライン制御はそれほど簡単ではない。その理由は、下記のように沢山の並行オペレーションが互いに関係し合っているからである。

- 対応するキャッシュ管理による命令フェッチ;
- 色々なアドレス指定方式によるアドレス生成;
- 独立のコンテンション解消によるクロスバススイッチアクセス;
- メモリ転送;
- プログラムカウンタロード/インCREMENTによるループアドレス比較;
- ループカウンタディCREMENT/リロード;
- ルーピング深さディCREMENT/ロード;
- マルチブライ;
- シフト;
- 加算/減算;
- 他のパラレルプロセッサとの同期;
- 割込み検出/優先順位指定。
- 線形コードの端的な実行において「異常」を生じさせるパイプライン「事象」は下記の通りである;
- 命令キャッシュミス;
- グローバルバスまたはローカルバスあるいはその双方におけるコンテンション;
- ループ;
- ブランチ及びコール;
- 割込み;
- アイドリング;

同期。

以下、これらの事象について図解的に説明する。以下の説明中「pc+1」及び「pc」という略号は、それぞれ、プログラムカウンタ3100が正常にインCREMENTされるか、されないかを示す。マーキングしてあるパイプライン境界は、スレーブクロックフェーズ及びこれに続くマスタークロックフェーズ、即ち1 s:m 1よりなるステージである。サイクルを何回でも繰り返すことができる場合は「1……1」で示してある。

10 キャッシュミスパイプラインシーケンス

第35図にキャッシュミス用のパイプラインシーケンスを示す。この場合、キャッシュミスはスレーブフェーズの間に検出され、これによってパラレルプロセッサの同期信号40がイナクティブになり、SIMD休止信号3007がアクティブになり、プログラムカウンタ3100はインCREMENTされず、パイプライン3105はロードされない。パイプラインは休止する。前の命令はそのままアドレスを生成し続けるが、アドレスレジスタ3202は変更されない。その前の命令はそのままデータユニットのオペレーションをリピートし続けるが、結果は記憶していない。しかしながら、クロスバススイッチアクセスは、ストアの場合はメモリに対して、またロードの場合は一時保持ラッチ3018及び3019に対してコンプリートに行われる。これらのアクセスは、それ以後の実行ステージの反復時には実行されない。

キャッシュミス処理要求信号3115は転送プロセッサ11へ送られる。パラレルプロセッサ100~103はその信号の受信応答 (アクリッジ) が得られるまで待機し、受信応答が得られたならばキャッシュミス情報を転送プロセッサ11へ転送する。そして、パラレルプロセッサ100~103は現フラグが転送プロセッサ11からの信号によってセットされるまで再度待機する。現フラグがいったんセットされると、同期信号40は再度アクティブになることができ、SIMD休止信号3007がイナクティブになって、命令フェッチング及びPC3100のインCREMENT動作を再開することができる。これによってアドレスユニット3001及びデータユニット3000はリリースされて、各々のオペレーションを完了する。また、一時保持ラッチ3018及び3019から各々の行き先レジスタへのロードがコンプリートに行われる。

キャッシュミス中に割込みが起こったとすると (これは定義によってPC3100のロード後2つの遅延スロット命令中には起こらない)、キャッシュミス処理要求信号3115をイナクティブにすることによりそのキャッシュミスは打ち切られる。これによって、その時必要ではないかも知れないコードがフェッチされるのを待つような無駄を防ぐ。転送プロセッサ11は、キャッシュミス処理要求信号3115がイナクティブになったことを検知すると、キャッシュミス処理の実行を中止する。

50 コンテンション解消パイプラインシーケンス

コンテンション解消のためのパイプラインシーケンスを第36図に示す。この例では、ローカルバス3006及びグローバルバス3005の両方でコンテンションが見られる。ここで、コンテンションは、2つ以上のパラレルプロセッサのローカルポート3006またはグローバルポート3005あるいはその両方が同じメモリ内にあるアドレスを同時に出力している状態と定義される。それらの対象はロードまたはストアあるいはその両方の如何なる組合せでもよい。コンテンションは、実行パイプラインステージの

ステップフェーズの間にクロスバスイッチまたは信号32 10及び3212によりローカルポート3006及びグローバルポート3005のロジックへ知らされる。パラレルプロセッサの同期信号40はイナクティブになり、SIMD休止信号3007はアクティブになる。

実行パイプラインステージがリビートされ、ポート30 05及び3006はアドレスパイプラインステージでアドレスユニットにラッチされたアドレスをそれぞれ再出力する。これが首尾よく行われると、メモリ10へストアがコンプリートし、一時保持ラッチへロードがコンプリート

する。実際は、ローラだけは第1ポートの保持ラッチ30 18及び3019に対して行われ、コンテンションを解消する。第2ポートは、ロードの場合、直接行き先レジスタの中へコンプリートすることができる。

この例では、ローカルバス3006は最初のリトライで成功する。それがストアの場合は、そのままメモリ10へ入り、ロードであれば、データは一時保持ラッチ3019へ書き込まれる。この例のグローバルバス3005は2回リトライしてからやっと転送に進事ができる。

リトライが実行される一方で命令フェッチングは終わっている。コンテンションが検出される前に次の命令がフェッチされたが、コンテンションが完全に解消されるまで実行は開始されない。次の命令が繰り返しフェッチされるが、パイプラインにはロードされない。

コンテンションが解消されると、同期信号40は再度アクティブになることができ、SIMD信号3007はイナクティブになり、命令フェッチングの再開が可能となる。

ループ制御パイプラインシーケンス

ループ制御のためのパイプラインシーケンスが第37図に示されている。この例では、ループは1つだけ定義されている(ループエンドレジスタ1 (3116) , ループカ 40
ウントレジスタ1 (3119) 及びループリロードレジスタ1 (3122) を用いる)。このループは2つの命令を含んでおり、ループ開始前のカウンタ値は2である。これらの原理は3つのすべてのループにも適用可能である。

図示の例では、プログラムカウンタ3100が(スレーブフェーズの間に) ループエンドレジスタ3116と内容が等しいことが判明すると、ループカウンタ3119は「1」と比較される。これは等しくないから、プログラムカウンタ3100はスタートアドレスレジスタ3111よりリロードされ、ループカウンタ3119は1だけディクリメントされ、

現ルーピング深さビット3108(ビット(22)及び(23))が最大ルーピング深さビット3108(ビット(24)及び(25))よりリロードされる(この例では、CLDビットは変化しない)。

次に、ループ再度リビートされるが、今度はループの終わりが検出され、ループカウンタ3119は「1」であるので、プログラムカウンタ3100は、スタートアドレスレジスタ3111よりロードされる代わりに、次の命令にインクレメントされる。ループカウンタ3119はループリロードレジスタ3122よりリロードされ、現ルーピング深さビット3108(ビット(22)及び(23))は1だけディクレメントされる。

ブランチまたはコールのためのパイプラインシーケンスは第38図に示されている。ブランチアドレスがプログラムカウンタ3100に書き込まれると、PC+1の値(スレーブフェーズ中に計算される)がRET3103にロードされる。これは2番目の遅延命令後の命令のアドレスであり、コールのリターンアドレスである。

ブランチアドレスは、レジスタから24ビットの即値であるいはプログラムカウンタ3100の現在値に24ビットのインデックスを加えることにより得ることができる。

ブランチアドレス及びリターンアドレスをセーブするについての問題は割込みが遅延スロット命令の間に許可された場合に生じる。このような問題を防ぐために、2つの遅延スロット命令のフェッチパイプラインステージの間割込みはロックアウトされる。それには、アドレスパイプラインステージのスレーブフェーズの間にプログラムカウンタ3100の行き先をデコードする必要がある。割込みのロックアウトは、2つの遅延スロット命令がフェッチされるまでは条件はテスト不可能であるため、条件付きブランチを伴う。

同期の所で説明したように、ブランチ及びコールは同期に関する限り1つの命令として扱われる。従って、同期信号40は2つの遅延スロット命令の間図示のタイミングでイナクティブになる。これは、条件に関わらず、条件付きブランチ及びコールについても同じである。

また、条件付きコールは条件付きブランチが入れられた場合にのみRET3103(リターンアドレス)をプッシュすることにより行われるから、SIMDでの条件付きコールは、「スレーブ」パラレルプロセッサ101~103はそのブランチが入れられているかどうかを知らないことから、問題が生じる可能性がある。そのため、これらのパラレルプロセッサ101~103はRET313をプッシュすべきかどうかを知らず、スタックコンシステンシーを招来する結果ともなる。この問題を解決するために、「マスター」SIMDパラレルプロセッサ100から「スレーブ」パラレルプロセッサ101~103へ「SIMDブランチ入り」という信号30 08が出力され、スレーブプロセッサはこの信号をPRET命令でRET 3103をプッシュすべきか否かを判断するために使用する。この信号は、図示のタイミングでアクティブ

になる（あるいはイナクティブ状態のままである）。

割込み

第39図に割込み用のパイプラインシーケンスが示されている。このシーケンスはMIMDまたはSIMDにおける何らかのハードウェア用のものであるが、割込みソースが「スレーブ」パラレルプロセッサ101～103であれば、このシーケンスは図示の如く、「スレーブ」PP・ツォ・「マスター」PP割込み信号3010によってキックオフされる。「スレーブ」パラレルプロセッサ101～103は、図示のように、「マスター」パラレルプロセッサ100が「マ

スター」・ツォ・「スレーブ」PP割込み信号3009を出力するまで待機する。
イネーブル化された割込みが検出されると、疑似命令のシーケンスが開始される。最初の命令は割込みベクトルアドレスを計算し、そのベクトルをプログラムカウンタ3100へフェッチし、プログラムカウンタの旧値（リターンアドレス）をRET3103にコピーする。2番目の命令はRET3103をプッシュする。3番目の命令はステータスレジスタ3108をプッシュし、そのS、I及びCLDビットをクリアする。

割込みルーティンの最初の2つの命令は、ステータスレジスタ3108がプッシュされ、そのS、I及びCLDビットがクリアされる前にフェッチされる。従って、これらのS、I及びCLDビットの機能は、ステータスレジスタ3108がプッシュされ、S、I及びCLDビットがクリアされるまで割込みロジックによってディセーブル化される。

アイドルパイプラインシーケンス

第40図にアイドル命令用のパイプラインシーケンスが示されている。アイドル命令は、そのアドレスパイプラインステージのスレーブフェーズの終わり以前にデコードされ、プログラムカウンタ3100がインCREMENTされるのを止める共に、パイプラインに次の命令がロードされるのを止める。MIMD休止信号はイナクティブになり、SIMD休止信号がアクティブ化される。命令フェッチングは、割込みロジックがイネーブル化された（許可された）割込みを検出するまで停止する。そのため、イネーブル化された割込みが検出されると疑似命令のシーケンスはキックオフされる。割込みソースが「スレーブ」SIMDパラレルプロセッサ101～103から発せられた場合、割込みシーケンスは、「マスター」・ツォ・「スレーブ」PP割込み信号3009がアクティブになるまでキックオフされない。

並列転送がアイドル命令でコードかされている場合は、それらの並列転送は割込みが起こった時割込みルーティンが実行される前に行われる。

同期

入力される同期信号が有効になるまで待機する同期MIMDまたはSIMD用のシーケンスが第41図に示されている。次の命令は、所望のすべてのパラレルプロセッサがアクティブな同期信号を出力するまで命令パイプラインには

フェッチされない。

アドレスユニット

アドレスユニット3001内のロジックは圧倒的にアドレスパイプラインステージで稼働して、実行ステージでクロスバスイッチが接続されたメモリ10のアクセスに必要なアドレスを計算する。しかしながら、実行ステージのメモリアccessは、2つのポート3005及び3006のクロスバスイッチコンテンションを独立に解決しなければならないので、これもやはりアドレスユニットの制御下にある。そのため、コンテンションの解消が行われている間はパイプラインを休止させるためにアドレスユニット3001からPFCユニット3002へフィードバックが用意されている。また、実行ステージの間にレジスタアクセスとアライナ/抽出器3003のオペレーションを実行する制御ロジックも設けられている。

第32図にアドレスユニット3001のブロック図を示す。この図から明らかなように、このユニットの主部は、同一構成の2つの16ビットサブユニット3200及び3201からなり、サブユニット3200はレジスタA0～A3 3202からアドレスを生成し、サブユニット3201はレジスタA4～A7 3207よりアドレスを生成する。これら2つのサブユニットは、それぞれ3200をグローバルサブユニット、3201をローカルサブユニットと称する。

ローカルサブユニット3201は幾分誤称と言えなくもない。と言うのは、単一メモリアccessが指定され、それが共用SIMDロードではない場合は、そのアクセスはサブユニット3200からでも3201からでも出すことができ、グローバルバス3005を介して行われることになるからである。そのために、マルチプレクサ3212～3214がこれらのサブユニットの外部に設けられている。2つの並列アクセスが指定された場合は、それらのアクセスはそれぞれ対応する名称のサブユニットより出される。共用SIMDモード（ローカルポート3006を介する）はローカルサブユニットを用いなければならない。

これらのサブユニット3200及び3201は16ビットアドレスで動作し且つ16ビットアドレスを生成するが、ユーザソフトウェアは、将来設計ではより大きなアドレスを生成し得るサブユニットを具備することも考えられるので、FFFFhから0000hへあるいはこの逆向きのローリングをベースとするものである。40

通常のパイプライン遅延は、命令により変更されるアドレスレジスタ3202及び3220、インデックスレジスタ3203及び3223、修飾子レジスタ3204及び3224あるいはモジュロレジスタ3205または3225は次の命令によって参照することができないという制約をユーザに強いる。これらのレジスタは次の次の命令によって参照することが可能である。そのため割込みが起こっても望ましくない結果を来すことはない。

グローバルサブユニット3200と3201はレジスタ番号以外

充分であろう。ただし、接続の仕方及び使い方には両者に若干の違いがあるので、これを重点的に説明するが、内部構造は両サブユニット共同である。

各サブユニット内には4つの16ビットアドレスレジスタ3202 (A0-A3) または (A4-A7) が具備されている。これらのレジスタには間接アドレスが入っており、その内容は不変のまま使用されるかまたはこれにインデックスが付加される。インデックスを付加する場合は、レジスタ3202の前の値を索引付け (インデキシング) により得られた値に置換するやり方を選んでよい。

アドレスレジスタ3202内の値は、転送されたデータサイズに関係なくバイトアドレスとみなされる。未整合ワードまたはハーフワードの転送は個々にコードできるがそれには2つの命令が必要である。それについては後に述べる。

パラレルプロセッサ100~103のアドレスアクセスはすべてアドレスレジスタ3202または3222をソースとしなければならない。オブコード内の即値アドレスをコード化する能力は具備されていない。これは、SIMDタスクが通常各パラレルプロセッサについて同一アドレスを指定したがることはないので、重要性は低いと考えられる。また、そのような能力は、MIMDアルゴリズムはどのパラレルプロセッサでもランできるように書かれるべきであるから、MIMDにとってもやはり重要性は低いと思われる。

アドレスレジスタA7 3227はスタックポインタとしてリザーブされている。このアドレスレジスタは他のアドレスレジスタ3202あるいは3222と同様に参照することができるが、レジスタA7の内容を調節する場合は、割込みがいつでも起こり得るので注意しなければならないことは明らかである。PUSH, POP及び割込みはプッシュを事前インCREMENTとして、またポップを事後インCREMENTとして扱う。

各サブユニット3200または3201内には16ビットの4つのインデックスレジスタ (X0-X3) 3203及び (X4-X7) 3223が具備されている。これらのインデックスレジスタの内容は、索引付アドレス指定を行うために、指定されたアドレスレジスタ3202または3222の内容に対して加算または減算するべくオブコードによって要求することができる。この加算/減算はアドレスがクロスバスイッチ20に送出される前または後に実行することにより、それぞれ事前または事後インデキシングを行うことができる。事前インデキシングにより生成されたアドレスはアドレスレジスタ3202または3222に戻してストアするやり方を選ぶこともできる。事後インデキシングについては必ずこのやり方によらなければならない。

オブコードによってアクセスが1つだけ指定された場合は、アドレスレジスタ3202または3222と同じサブユニット内の4つのインデックスレジスタ3203または3223の中の1つ (例えばA0とX2, A6とX4, ...) をインデックスソースとして指定することができる。指定可能なインデキ

シングモードは事前または事後、加算または減算で、アドレスレジスタ3202または3222の変更を伴う場合と伴わない場合とがある。

2つの並列アクセスが指定された場合は、アドレスレジスタ3202または3222と同じ接尾辞を有するインデックスレジスタ3203または3223が用いられ (例えばA2とX2, A5とX5)、事後-加算インデキシングのみが可能である。

インデックスレジスタ3203及び3223の内容値は、転送中のデータサイズに関わらず、常にバイトアドレスとして解釈される。

インデックスレジスタインデキシングの代替的インデキシング方法としては、短即値インデキシングまたは暗黙即値インデキシングがある。短即値インデキシングはアクセスが1つだけ指定された場合に可能で、インデックスとして3ビットの短即値を使用することができる。インデックスレジスタインデキシングのモードは事前または事後、加算または減算でアドレスレジスタ3202または3222の変更を伴う場合と伴わない場合がある。

2つの並列アクセスがコード化されている場合は、事後インデキシングによる+1の暗黙即値、及び事前インデキシングによる-1の暗黙即値だけが指定可能である。これらを指定すると、2つの並行アクセスが並列転送がコード化されていても8, 16または32ビットのスタックをアクセスすることができる。

短即値または暗黙即値インデキシングを指定する場合、即値は指定されたワードサイズが8, 16または32ビットの中の何れであるかによってシフト3208又は3228により左へ0, 1または2ビットシフトしてからアドレスレジスタ3202または3222より読み出された値に加えられる。従って、短即値インデックスは0~7「単位」であり、暗黙即値インデックスは+/-1「単位」である。ただし、ここで「単位」はデータサイズである。アドレスレジスタは常にバイトアドレスを有するため、シフトされない。

各アドレスレジスタ (A0-A3) 3202または (A4-A7) 3222には8ビットのアドレス修飾子レジスタ (Q0-Q3) 3206または (Q4-Q7) 3224が具備されている。これらの修飾子レジスタは、オブコードに組み入れることのできないアクセスに必要な補助的情報を記憶する。この種的情報は、通常、サイクル単位ベースで変更する必要はない。

レジスタA7 3227はスタックポインタとして用いるよう割当てられているため、レジスタQ7 3229のビット6~0はそれぞれハードワイヤードにより0000010に接続されている。以下、Qレジスタ3204及び3224の個々のビットの機能について説明する。

パラレルプロセッサのアドレス空間は2つの半部に分けられる。即ち、データ空間 (クロスバスイッチが接続されたメモリ10) 及びI/O空間 (パラメータRAM、メッセ

レジスタ及びセマフォフラグ)である。このビットが「1」の場合、アクセスはI/O空間に対してなされる。このビットの「0」はクロスバスイッチが接続されたRAM10へのアクセスを指示する。

2の累乗モジュロビットが「1」になっていると、そのビットはサブユニット3200または3201対応するモジュロレジスタM0 3205またはM4 3225中の「1」(「1」が複数の場合もある)により指示された位置でアドレス加算器3206または3226上のキャリーパスをブレイクしたいということを示している。これによって、2の累乗次元マトリクスアドレス指定を行うことができる。このビットが「0」の場合、アドレス加算器3206または3226は普通の16ビット加算器/減算器として動作する。

逆キャリーアドレス指定ビットが「1」にセットされていると、逆キャリーアドレス指定がイネーブル化される。これはアドレス加算器/減算器3206または3226のキャリーパスの向きを逆転させる。2の累乗インデックス(例えば8, 16, 32等)により索引付アドレス指定方式を指定する場合、これにはFFTやDCTで必要のようにして計数するという作用がある。このビットが「0」であると、アドレス加算器3206または3226は普通の16ビット加算器/減算器として動作する。

共用SIMDロードビットは、「1」にセットされると、ロードが指定される場合それは共用SIMDロードであるべきであるということを指示する。このビットは、共用SIMDロードの性格の故に、「マスター」SIMDパラレルプロセッサ100がロードを指定する時、このプロセッサのQ4-Q6 3224だけに関係する。これは、パラレルプロセッサのローカルバス3006を、ロードの持続時間中、直列接続させる。このビットが「0」であると、共用SIMDロード機能はディスエーブル化される。このビットを「スレーブ」パラレルプロセッサ100~103、あるいは「マスター」SIMDパラレルプロセッサのQ4-Q6以外でセットしても何ら効果はない。ストアはこのビット値によって左右されない。符号拡張ビットが「1」にセットされると、ハーフワードまたはバイトのロードは、ビット15またはビット7をパラレルプロセッサのレジスタにロードされた時すべての再上位ビットにコピーさせる。これはアライナ/抽出器の機能である。このビットが「0」の場合、再上位ビットにはすべて「0」が入れられる。

2つのサイズビットは転送されるデータのサイズを指定する。コーディングの内容は、00--8ビット、01--16ビット、10--32ビット、11--逆向きとする。これらのビットはアライナ/抽出器3003、ストアに対するバイトストローブ、及び符号拡張機能を制御する。

アドレス演算論理装置(ALU) 3206及び3226は、キャリーパスの方向を逆転することができるあるいはキャリーパスをブレイクすることができるという点以外は、通常の16ビット加算器/減算器である。

インプレースFFTを行う場合、ソースデータあるいは

結果のアドレスはアクセス困難となるほどスクランブルされる。しかしながら、この場合のスクランプリングには、アドレス加算器3206または3226のキャリーパスの向きを逆にすればかなり容易にスクランブル状態から回復する(アンスクランプリング)ことができるというような秩序がある。DSP共通のこの特徴は通常逆キャリーアドレス指定方式またはビット逆転アドレス指定方式と呼ばれる。

FFTポイント数を2で割った数の2の累乗数に等しい2の累乗インデックス(例えば8, 16, 32, ...)が逆キャリーリップルパスを用いてアドレスレジスタ3202または3222からのアドレスに加算される。その結果の値はアドレスとして用いられ、アドレスレジスタ3202または3222に記憶される。これによってデータをアンスクランブルするのに必要なアドレスのシーケンスが生成される。例えば、インデックスが8で、アドレスレジスタの初期値が0であるとする、0, 8, 4, C, 2, A, 6, E, 1, 9, 5, D, 3, B, 7, Fのシーケンスが生成される。

逆キャリーの特徴は、2の累乗以外のどのようなインデックスについても用いるが、有用な結果がもたらされるとは限らない。この特徴は、指定されたAレジスタに対応するQレジスタ3204または3224の逆キャリービット「1」にセットされている場合のみ有効となる。

クロスバメモリ10の周りにデータを分散させる時は、連続データをアクセスし、境界条件を処理し、あるいはアレイされたデータをアドレスするために、特定の次元で「ラップアラウンド」が必要になる状況が充分起こり得る。これを容易にサポートするために、アドレス加算器3206または3226のキャリーパスを1つまたは2つ以上の選択された場所でブレイクする能力が具備されている。

このようなブレイクの場所はモジュロレジスタM0 3205またはM4 3225によって決定される。モジュロレジスタのビットnに「1」が入っていると、アドレス加算器のビットn-1とnの間でキャリーパスがブレイクされる。これによると、2nモジュロバッファが実施可能である。モジュロレジスタ3205または3225には所望通りに幾つでも「1」をプログラムによって入れることができる。これによって、各次元が2の累乗の係数値であるとして、多次元アレイを実施することができる。

この特徴は、指定されたアドレスレジスタ3202または3222に対応する修飾子レジスタ3204または3224の2の累乗のモジュロビットが「1」にセットされている時のみアクティブとなる。その他の場合は、通常の線形アドレス指定方式が適用される。

ローカルポート及びグローバルポート

グローバルポート3005及びローカルポート3006の主たる特徴はアライナ/抽出器3003である。アライナ/抽出器は8ビット、16ビット及び32ビットデータ、符号拡張、非整合アクセス及び共用SIMDロードを扱う。これら

10

20

30

40

50

の機能を果たすため、アライナ/抽出器3003は、基本的には、必要なオペレーションを得るために接続された一群のマルチプレクサよりなる。グローバルポート3005及びローカルポート3006は各々独立に動作し、従って、どちらか一方についてした説明は他方のポートについても当てはまる。ただし、共用SIMDロードは例外で、これについては以下に説明する。

ロードまたはストアのデータサイズは、指定されたアドレスレジスタ3202または3222に対応する修飾子レジスタ3204または3224内で定義される。有効に選択し得るデータサイズのビット数は8, 16または32ビットである。このように、データサイズは、どのアドレスレジスタ3202または3220がアクセスされているか及び修飾子レジスタ3204または3224内の値によってサイクル単位ベースで変化し得る。

メモリ10とパラレルプロセッサ100~103との間のクロスバスイッチを通しては、たとえ指定ワードサイズが8ビット16ビットであっても常にフル32ビットのワードが転送される。8ビットまたは16ビットの量のロードを行う場合は、アドレス及びワードサイズの最下位 (LS) ビットによって32ビットより適宜のバイトが抽出される。このように抽出されたバイトは、必要に応じて右シフトし、右寄せしてから行き先のパラレルプロセッサレジスタに入れる。上位のバイトはすべて0を入れ、修飾子レジスタ3204または3222で符号拡張が指定されていれば、再上位バイトに再上位 (MS) ビット (15または7) をコピーする。

クロスバスイッチが接続されたメモリ10に8ビットまたは16ビットの量を書き込む場合は、データ (右寄せされている) はアライナ/抽出器3003によって4回または2回繰り返し、32ビットワードを作る。このワードは、次いで、アドレス及びデータサイズのLSビットによりセットされる4バイトのストロブと共にクロスバ20を介して書き込まれる。このようにして適宜のバイトがメモリに書き込まれる。

上記のデータのロード及びストアの説明では、アドレスは位置合わせされているものと仮定してある。即ち16ビットのアクセスはLSビット=0のアドレスに対してなされ、また32ビットのアクセスは2つのLSビット=00のアドレスに対して行われる。(8ビットの量は常に位置合わせされる)。

それでも、位置合わせされていない16または32ビットのデータをもアクセスすることができるようになっている。これは自動的に行われるのではなく、ユーザがデータの上位側部分及び下位側部分のロードまたはストアを具体的にエンコードする必要がある。その結果、データの「上位側路」、「下位側ロード」、「上位側ストア」及び「下位側ストア」の4つの命令が得られる。これらの命令は、バイトアドレス及びデータサイズを用いてアライナ/抽出器3003を制御し、ロードの場合は、行き先

レジスタの適宜の部分をロードするだけである。そのためにはレジスタは各個にバイト書き込み信号を持つ必要がある。このような理由から、位置合わせされていないロードはデータレジスタ3200だけに限定される。

実際は、「下位側ロード」及び「上位側ストア」という命令は通常のロード命令及びストア命令である。これに「上位側」の等価オペレーションが続く (または先行する) 場合は、何も転送されない。アドレスが位置合わせされていない場合は、適宜のバイトのみがメモリにストアされるかまたはレジスタにロードされる。

説明に資するため、位置合わせなしのオペレーションの例を第42図及び第43図に示す。これらはいずれも余計とも言えるような自明の例である。

共用SIMDロード

コンボリユーション等においては、各ハードウェア、各サイクル毎に2つのアクセスを並列に実行することがひつような場合が時々ある。その例が、例えばデータがクロスバスイッチが接続されたメモリ10の中のいずれかの場所からグローバルポート3005を介して取り出されるような場合であり、あるいはカーネル値のように各パラレルプロセッサ100~103に「共通した」情報の場合である。後者のような場合は、どちらかと言うと、ローカルポート3006を介して供給されるものと考えられる。このような情報を1つのデータソースから同時にローカルポート3006へ伝送するために、ローカルクロスバ6を直列接続する単方向バッファが設けられている。

これらの直列接続は、ローカルアドレスサブユニット3201のアドレスレジスタ3222が、共用SIMDロードビットに対応する修飾子レジスタ3224でセットされてアクセスされ、ロードが指定された時、SIMDモードでのみ行われる。その他の条件下ではすべてローカルデータバス6は互いに切り離される。この直列接続がなされると、パラレルプロセッサ1-3 101~103 (「スレーブ」SIMDパラレルプロセッサ) はクロスバスイッチ20によって無視される。

直列接続バッファは単方向性であるから、共通データは「マスター」SIMDパラレルプロセッサPP0 100. に対向する4つのクロスバRAM 10. 0, 10. 2, 10. 3及び10. 6 (即ちアドレス範囲0000h~1FFFh) だけに記憶することができる。

コンテンション解消

コンテンション解消の目的は、偶然 (あるいは意図的に) システム内の何れか2つのデバイスによって同一RAMを同時にアクセスしてしまうのではないかという心配からユーザを解放することにある。各クロスバRAMには7本のバスが接続されている。従って、常にコンテンションを回避することに気を配るのはかなりの拘束的要因である。

SIMDモードでは、すべてのパラレルプロセッサ100~103はコンテンションが解消されるまで待機する必要がある

る。そのために、「SIMD休止」信号3007がパラレルプロセッサ100~103の間に巡回伝送されており、この信号はコンテンションが解消されるまで、どのパラレルプロセッサによってもアクティブ化することができる。この信号は同期信号40の線路を介して送られる。

クロスバアクセスは、グローバルポート3005及びローカルポート3006がアクセスしようとしているRAMの所有権をグラントされる（認められる）と同時に完了する。ストアの場合、アクセスが可能になると同時にメモリ10に対してコンプリートに行われる。ロードの場合は、パラレルプロセッサがすぐに実行を再開することができないと（他のポートでコンテンションが続いている、SIMD休止信号3007が依然としてアクティブである、同期したMIMDパラレルプロセッサが他のパラレルプロセッサに対して待機中である、あるいはキャッシュミスが起こった等の理由で）、ロードは実行が再開されるまで保持ラッチ3018及び3019へコンプリートされる。これは、データユニットのオペレーションもホールドされており、そのソースデータ（即ちデータレジスタ3300の内容）はストアによって上書きすることができないからである。同様に、ロードとストアが同一データレジスタに対してアクセスしており、かつストアがコンテンションにより遅延した場合は、ロードデータをラッチ3018または3019に一時的に保持しなければならない。

データユニット

データユニット3000内のロジックは専ら実行パイプラインステータスの間に稼働する。そのオペレーションはすべてレジスタのみかまたは即値とレジスタを使用する。間接（メモリ）オペランドはサポートされない。従って、メモリに対するデータ転送はストア及びロードとして具体的にコード化される。

データユニット3000のブロック図を第33図に示す。

このデータユニットの主構成要素は8のデータレジスタ3000、1つのフルバレルシフト3301、32ビットALU 3302、シングルサイクル16×16マルチプライヤ3304、論理値「1」を扱う特殊ハードウェア3303、一連のマルチプレクサ3305~3309等である。

また、バレルシフト3301及びALU3302との密接な関係の下に2つのレジスタ3310または3311が設けられている。2つのレジスタは一定の命令が実行される時これらのデバイス及びオペレーションを制御する。

データユニット3000内には8つのデータレジスタ

(D) 3300がある。これらは汎用の32ビットデータレジスタで、マルチポートになっているため大量の並列処理が可能である。ALU3302及びマルチプライヤ3304に対しては、メモリへとメモリからの2つの転送が行われている間に、それらの転送と同時に4つのソースを提供することができる。

マルチプライヤ（乗算器）3306は単サイクルハードウェアの16×16マルチプライヤである。32ビットの演算結

果はレジスタファイル3300へ戻される。ハードウェアは符号付及び符号なしの演算をサポートする。

第33図から明らかなように、データユニット内3000内には色々なハードウェアにデータを供給する数個のマルチプレクサが具備されている。ALU3302にフィードする2つのマルチプレクサ3306または3307（バレルシフト(BS) 3301を介して「1」を供給する）は個々のバイトの多重化をサポートするという点で若干異なっている。この特徴は「マージ多重 (MRGM)」という命令を実行可能にするためである。この命令は、各ソースの個々のバイトをオールゼロバイトと多重化するためにMFLAGSレジスタの4つ、2つまたは1つの最下位ビットを使用し、ALUの一方の入力に供給されるバイトがMFLAGSに従い混合されたsrc1バイトと00hバイトになるようにする。ALUの他方の入力には00hバイトとsrc2バイトの逆の組合せが供給される。すると、ALU3302はADD（加算）及びOR（論理和）演算を行って、src1からの一部のバイトとsrc2からの一部のバイトからなる結果を出力する。このオペレーションは、飽和处理、色膨張及び色比較、最大最小、透明度処理及びマスキングを行う上において非常に役に立つ。

バレルシフト3301はALU 3302の「反転」入力に接続されている。これによって、オプションズ (OPTIONS) レジスタ3310にセットアップされた定義済みのシフト（けた移動）量を用いてシフトと加算のオペレーションまたはシフトと減算のオペレーションを行うことが可能となる。これは、マルチプライヤは結果スケラを持たないので、非常に重要である。バレルシフト3301は0乃至31ビット位置（けた）だけ左または右にけた移動するができ、また0乃至31ビットのローテーションを行うことができる。

32ビットのALU 3302はありとあらゆる論理演算、加算及び減算を行うことができる。一部の命令はALU 3302を加算または減算のために2つのハーフワードまたは4バイトにスプリットさせることができるので、ALUはこれにより多重ピクセルに対するオペレーションを実行することができる。「1」ロジック3303は3つの異なるオペレーションをおこなう。即ち、左端「1」検出、右端「1」検出とワード内の「1」を計数するオペレーションである。これらのオペレーションはまとめてデータ圧縮、データ分割及び相關付等において色々な形で利用することができる。

ALU 3302の出力は1ビット左シフトを有し、これは除算反復ステップで用いられる。このシフトはオリジナルソースを選択してゼロインサーションにより左に1けた移動させるか、または2つのソースの減算結果を選んで1ビット左に移動させ、「1」を挿入する。

「多重」フラグズレジスタ (MFLAGS) レジスタ3311は32ビットレジスタで、「add multiple（多重加算; ADD M）」、「subtract multip」（多重減算; SUBM）」また

は「compare multiple (多重比較CMPM)」命令の結果を収集するのに用いられる。ALU 3302はオプションズレジスタ3310のALUビットの値によって4つ、2つまたは1つのピースにスプリットすることができる。「多重」フラグズレジスタ3311の4つ、2つまたは1つの最上位ビットは、上記3つの命令のキャリー、ボローまたはイクエート(相等化)ビットによってロードされる。

オプションズレジスタ3310は、2つの制御フィールド、「多重」命令と共に用いるALUSプリットビット、及びシフトと加算の命令並びにシフトと減算の命令のためのパレルシフタの定義済量を持っている。

オプションズレジスタ3310の3つのALUビットは、ALU 3302を各々2, 4, 8, 16及び32ビットサイズのピースにスプリットすることを可能にする。そのためのコーディングの指定内容は:000—2ビット、001—4ビット、010—8ビット、011—16ビット、100—32ビットである。ただし、この実施例では、8ビット、16ビット及び32ビットだけが可能である。これらのビット値はADD M, SUBM, MRGM及びCMPM命令のオペレーションを制御する。

マージ多重命令

第44図は、第33図のハードウェアを用いてMFLAGSレジスタをセットさせるスプリット可能ALU命令とマージ多重(MRGM)命令との組合せにより実行可能な複雑なオペレーションをいくつか示している。これらの例は、データ操作部分のみを示し、一般にはこれらのオペレーションを多数含むループになる。

第44図の飽和加算の例では、ADDM命令は4つの8ビット加算を並行して行い、各8ビット加算の間にキャリーアウト(オーバーフローを示す)が起こっているかどうかによってMFLAGSレジスタをセットさせる。16進67即ち67hをEFhに、またCDhを45hに加える8ビット加算は、どちらも8ビット値のキャリーアウトを生じ、その結果MFLAGビット0及び1がセットされ(MFLAGSレジスタの4つの最下位ビット(最下位ビットから4番目までのビット)のみがMRGM命令にとっては有意であるということに注意)、MFLAGSレジスタは「3」にセットされる。D3は前にFFFFFFFhにセットされているから、MFLAGSレジスタの値はD2に入っている前のオペレーションの結果かまたはD3に入っている16進「FF」の飽和値のどちらかを選択するのに用いられる。

「最大」機能はSUBMの実行後MRGM命令によって同じ2つのレジスタを用いることによって得られる。SUBMは、並行する4つの8ビット減算の結果一方のレジスタの中の32ビット値の8ビットが他方のレジスタの対応する8ビットより大きくなっているか否かによってMFLAGSレジスタのビットをセットする。この例から明らかなように、MFLAGSレジスタに「5」(または4つの最下位ビットが2進形式で「0101」)の結果が入っているということとは、16進「EE」が16進「67」より大きかったというこ

と及び16進「AB」が16進「23」より大きかったということを示す。MRGM命令によってMFLAGSレジスタ中の結果を用いることにより、レジスタD0とD1に入っている対応する値の中大きい方の値が最終結果としてD2に記憶される。

透明度処理については、「透明色」か後で図示のような8ビット値の書き込みを保護する保護色値かの比較が行われる(図示例で値「23」は透明を示す)。CMPM命令は並行して4つの8ビット比較を行い、同じ比較に基づき互に対応する4つのMFLAGビットをセットする。図示の例では、右から3番目の比較結果だけが「等しく」、このことはMFLAGSレジスタに「4」(2進形式で「0100」)が入ることにより指示される。すると、MRGM命令は、右から3番目の8ビットを除き、結果についてはD0の値のみを使用することになる。

色膨張は2進マップ中の論理値「1」または「0」に基づく2つの多重ビット値の選択を含む。図示の例では、16進「6」(2進形式0110)の4ビット値がMFLAGSレジスタに移動される。この例のMRGM命令は、MFLAGSレジスタの対応する記憶場所の値に従い単にD0及びD1の8ビット値のどちらかを選択することである。

色圧縮においては、対応する値が各特定の色値と一致するか否かに基づき2進マップが作成される。図示例の場合、MFLAGSレジスタ中のCMPM命令の結果は所期の結果であったということを示している。

ガイドイドコピーの例では、2進パターンアレイを用いてソースのどの値の行き先にコピーするかを決定する。図示例では、D0の2つの上位8ビット値がD0へコピーされる。

以上の例では例示説明のため8ビットのデータ値を用いたが、データ値の数並びにサイズは8ビット値4つに限定されるものではない。

ここでは、マージ命令と共に使用される演算用多重命令の重要な組合せを例示説明したが、このシステムではこれ以外の多くの組合せや有用なオペレーションが可能である。また、この発明のシステムによれば、マスキレジスタをセットさせる演算用多重命令を用い、続けてマージ命令を実行することによって多数の有用なオペレーションを得ることができるということも重要である。

オプションズレジスタ3310の2つのOPTビットは、シフトと加算の命令及びシフトと減算の命令の間にパレルシフタ3301が実行するシフトの形を指定する。そのコーディングの指定内容は次の通りである:00—論理右シフト(右けた送り)、01—算術右シフト、10—論理左シフト、11—ローテーションである。

オプションズレジスタ3310のAMOUNTビットは、シフトと加算の命令またはシフトと減算の命令が実行される場合に生じる上記OPTビットで指示される形のシフトまたはローテーションのビット数を指示する。

命令

次頁以後にパラレルプロセッサ100～103で使用可能な命令をいくつか詳細に例示説明する。これらの命令でドット (.) は所望の通りに割り当てることができるオペレーションコードを表す。これらの中一部の命令は既に説明したものである。

例示の順序は次の通りである。

シンタクス

LCK

*

オペレーション MIMD PP の同期待ち

オペランド なし、並列転送による

コード化形式 31 27 22 19 16 0

0000	000	000	並列転送	-
------	------	-----	-----	------	---

説明

この命令はMIMDの同期PPコードを開始するのに用いられる。この命令は同期レジスタの「1」で指示されたすべてのパラレルプロセッサ互いに同期するまでパラレルプロセッサを待機させる。次いで、次の命令が他のMIMDパラレルプロセッサと同期してフェッチされる。(アドレス及び実行パイプラインステージの実行は連続する各命令が同期してフェッチされるに従って行われる)。ULCKは同期コード実行を終了させる。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 影響なし

M ビット 影響なし

*① データユニット命令 (並列転送を伴うもの、伴わないもの) 及びシングルのオペレーション命令 (即ち並列オペレーションなし) ;

② データユニットのオペレーションと並列に行われる転送。

コード化形式

31	27	232221	18	1615	1110	8	7	5	2	0	
データユニットオペレーション			1	mdg	mdl	0	A47	0	A03	Dg	D1

説明 (Dレジスタ) データユニットのアドレスはD
グローバルサブユニットのアドレスはD
ら生成された間接アドレス。スタ
dsl1がロードされる。スタ
ットのアドレスタスインタフェース
ドレス1だけポストインデックス
間接、及びXだけのポストインデ
メント、2本のレジスタはD
定使用したindexレジスタはD
サブスクリプトを有する。dsl
ならない。

ステータスピット N - 影響なし

C - 影響なし

V - 影響なし

7. - 影響なし

2 ビット 影響なし

87
シンタクス $\parallel LD *An(mode), dst1 \parallel ST src2, *An(mode).$
オペレーション $*src1 \rightarrow dst1 \parallel src2 \rightarrow *dst2$
オペランド
A0-A3(mode {g } , Dn Dn, A4-A7(mode)) データユニット
ト並行オペレーションあり、または
アドレス指定モード 1だけポストインクリメント、ア
ドレスレジスタ修正
1だけプレデクレリメント、アド
レスレジスタ修正
A レジスタに関連するindex レジ
スタによるポストインクリメント
、アドレスレジスタ修正
間接、インデキシングなし

コード化形式

31	27	23	22	21	18	16	15	11	10	7	5	2	0
データユニットオペレーション					1	mdg	mdl	0	A47	1	A03	Dg	Dl

説明
(Dレジスタ) データユニットオペレーションと並行に、
グローバルサブユニットのアドレスレジスタ(A0-A3) か
ら生成された間接アドレスよりグローバルバスを介して
dst1がロードされる。これと並行に、ローカルサブユニ
ットのアドレスレジスタ(A4-A7) から生成された間接ア
ドレスよりローカルバスを介してsrc2がストアされる。
間接、1 だけポストインクリメント、1 だけプレデクレ
メント及びX だけポストインクリメント等のアドレス指
定モードは2本のバス上で独立にサポートされる。ここ
で使用したindex レジスタ はアドレスレジスタと同じ
サブスクリプトを有する。dst1及びsrc2は0 レジスタで
なければならない。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 影響なし

M ビット 影響なし

シンタクス ⁸⁹ `NUM1 src, dst`

オペレーション `src` の「1」の数 → `dst`

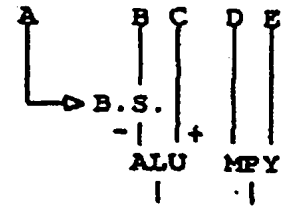
オペランド

`D, 0` 並列転送による

`any, any` 並列転送なし

ルーティング

A	B	C	D	E
0	src	0	-	-



コード化形式

31	27	22	19	16	11	8	5	2	0
0 0 0 0	...	src	dst	並 列 転 送					
0 0 0 0	...	src	dst	0 0 0 0 0	scde	dcde	0 0 0	0 0 0	0

説明 `src` 内の「1」が計数され、`dst` へロードされる。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 1(`src` 内容が全部0 の場合)、0(その他の場合)

M ビット 影響なし

91
シンタクス $\parallel ST\ src1, *An(mode) \parallel LD\ *An(mode), dst2$

オペレーション $src1 \rightarrow *dst1 \parallel *src2 \rightarrow dst2$

オペランド
Dn, A0-A3(mode {g}), A4-A7(mode), Dn) データユニット
並行オペレーションあり、または

アドレス指定モード 1 だけポストインクリメント、アドレスレジスタ修正
1 だけプレデクレリメント、アドレスレジスタ修正
A レジスタに関連する index レジスタによるポストインクリメント、アドレスレジスタ修正
間接、インデキシングなし

92

コード化形式

31	27	23	22	21	18	16	15	11	10	8	5	2	0
データユニットオペレーション					1	mdg	mdl	1	A47	0	A03	Dg	D1

説明

(Dレジスタ) データユニットオペレーションと並行に、グローバルサブユニットのアドレスレジスタ(A0-A3) から生成された間接アドレスへグローバルバスを介して $src1$ がロードされる。これと並行に、ローカルサブユニットのアドレスレジスタ(A4-A7) から生成された間接アドレスよりローカルバスを介して $dst2$ がロードされる。間接、1 だけポストインクリメント、1 だけプレデクレメント及びX だけポストインクリメント等のアドレス指定モードは2本のバス上で独立にサポートされる。ここで使用した index レジスタはアドレスレジスタと同じサブスクリプトを有する。 $src1$ 及び $dst2$ は D レジスタでなければならない。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 影響なし

M ビット 影響なし

93	シンタクス	ST src1, *An(mode) ST src2, *An(mode),	94
	オペレーション	src1 → *dst1 src2 → *dst2	
	オペランド	Dn, A0-A3(mode (g)) Dn, A4-A7(mode)) データユニット 並行オペレーションあり、または	
	アドレス指定モード	1 だけポストインクリメント、アドレスレジスタ修正 1 だけプレデクレリメント、アドレスレジスタ修正 A レジスタに関連する index レジスタによるポストインクリメント、アドレスレジスタ修正 間接、インデキシングなし	

コード化形式

31	27	23	22	21	18	16	15	11	10	8	5	2	0
データユニットオペレーション					1	mdg	mdl	1	A47	0	A03	dg	dl

説明

(0 レジスタ) データユニットオペレーションと並行に、グローバルサブユニットのアドレスレジスタ(A0-A3) から生成された間接アドレスへグローバルバスを介して src1 がロードされる。これと並行に、ローカルサブユニットのアドレスレジスタ(A4-A7) から生成された間接アドレスへローカルバスを介して src2 がストアされる。間接、1 だけポストインクリメント、1 だけプレデクレメント及びX だけポストインクリメント等のアドレス指定モードは2本のバス上で独立にサポートされる。ここで使用した index レジスタはアドレスレジスタと同じサブスクリプトを有する。src1 及び src2 は 30 レジスタでなければならない。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 影響なし

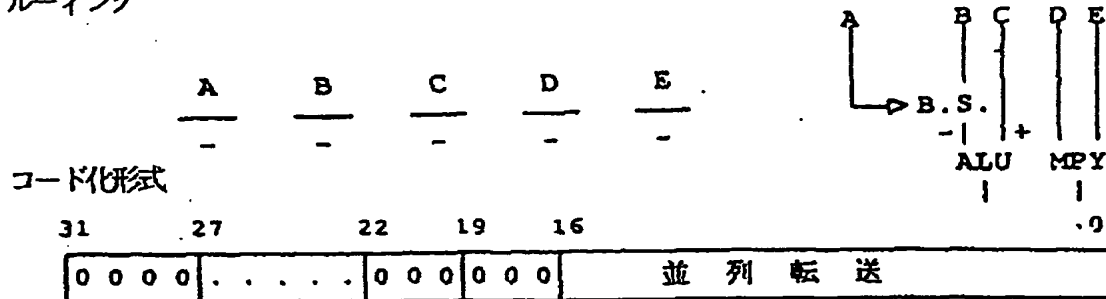
M ビット 影響なし

シンタクス 95
ULCK

オペレーション MIMD PP を互いにアンロックする

オペランド
なし 並列転送による

ルーティング



説明 この命令はMIMD PP を互いにアンロックする。これらのMIMD PP はその後次の命令フェッチによって独立の命令実行を再開する。

ステータスビット N - 影響なし

C - 影響なし

V - 影響なし

Z - 影響なし

M ビット 影響なし

転送プロセッサ

転送プロセッサ11はシステムメモリ10とシステム外部との間のインターフェースであり、特に外部メモリ15へのあらゆるアクセスに関する責任を負う。

転送プロセッサ11は、第57図に詳細に示すように、主としてメモリの1つのエリアと別のエリアとの間のブロック転送を行う。「ソース」及び「行き先」メモリはオンチップでもオフチップでもよく、データ転送はバス5700及びFIFOバッファメモリ5701を介して行われる。オンチップメモリはクロスバデータメモリ10、パラレルプロセッサの命令キャッシュ10、マスタープロセッサ命令キャッシュ14、及びマスタープロセッサデータキャッシュ13を含む（第1図及び第2図参照）。データメモリ10及びデータキャッシュ13はどちらも読み出し/書き込み可能である。命令キャッシュ14は書き込み専用である。

キャッシュを含むすべてのオペレーションはそれらのキャッシュに関連するロジックによって自動的に要求される。この場合、移動されるデータの量はキャッシュの「ライン」サイズであり、データは適切なセグメントレジスタにより指定された外部メモリ15とキャッシュのセグメントとの間で移動される。

クロスバデータメモリ10を含む転送は、パラレルプロ

セッサ100~103またはマスタープロセッサ12からの「パケット要求」に応動して実行され、バス5707を介して行われる。パケット要求は、移動されるデータ量を含む沢山のパラメータ及びソースアドレスならびに行き先アドレスによって転送を指定する。

ブロック転送

パケット要求はメモリの1つのエリアから別のエリアへの一般化された転送を指定する。ソースアドレスジェネレータ5704及び行き先アドレスジェネレータ5705は共に同じように記述されている。「ブロック」はデータアイテム（バイト、ハーフワード、ワードまたは長ワード）を単に線状に隣接させて次々に並べたもので良く、あるいはそのようなデータアイテムを並べた領域の複数個で構成したものでも良い。このアドレス指定機構によれば、最大3次元までの「アレイ」を指定することができる。そのため、単一のパケット要求により多数の2次元パッチを処理することが可能である。

最も内側の次元に沿ったデータアイテムは常に1単位離れている。より高次元のアイテム間の距離は任意である。

各次元のカウント数はソースアレイ及び行き先アレイ共同である。

30

40

50

第45図は単一パケット要求で指定することのできる複雑な形式のブロックの一例を示す。図示のブロックは、各々512の隣接ピクセルよりなる3本の線群が2群で構成されている。このようなブロックが必要となるのは、例えば、各々1つの線群について処理中の2つのパラレルプロセッサが3×3のコンボリューションを実行しようとするような場合である。

ブロックは第45図に示す如く下記のパラメータによって指定される。

ランレングス	隣接データアイテムの数、例えば 10
512ピクセル	
レベル2カウント	群中の「線」の数、例えば 3
レベル3カウント	「ブロック」中の「群」の数、例えば 2
スタートアドレス	ブロックの始めのリニアアドレス、例えば「SA」で指示されるピクセルのアドレス
レベル2ステップ	第1レベル群間の距離、例えばピクセル「B」と「A」のアドレスの差
レベル3ステップ	第2レベル群間の距離、例えばピクセル「D」と「C」のアドレスの差
VRAM補助	

次に、マルチプロセッサシステムと共にビデオRAMを使用するやり方に付いて第58図を参照しつつ説明する。図示例では、ビデオカメラからのCCD入力またはその他のビデオ信号入力がAD変換器5802によってクロック同期され、シフトレジスタ(S/R) 5801へ供給される。データは、シフトレジスタ5801へシフトインされ、そこからランダムメモリマトリクス5800へシフトアウトされるが、この例では、ランダムメモリマトリクス5800は第1図のメモリ15がそのまま使われている。Sクロック入力、シフトレジスタ5801に対する情報のシフトイン、シフトアウトを制御するのに用いられる。ランダムメモリマトリクス5800からのデータ出力は前述した如くパラレルプロセッサによって制御される。即ち、パラレルプロセッサは、画像処理、画像制御または図形識別を行うため、あるいはペーパーコピーその他のコピーから斑点等を除去するクリーニングを行うために情報を並列でも直列でも用いることができるようにデータ出力を制御する。ISP(イメージシステムプロセッサ)は第58図のポート21を介してビデオRAMのデータにアクセスする。シフトレジスタ5801とランダムメモリマトリクスとの相互作用の目的は、情報をプロセッサの動作速度に関係なく外部から非同期で取り込み、ランダムメモリマトリクスにロードすることができるようにすることである。このように情報が取り込まれたならば、転送プロセッサは前に述べたようにして情報の転送を開始する。入力情報には、一般に、水平同期信号、ブランキング信号及び垂直リフレッシュ信号等を含むNTSC標準信号が含まれており、これらの信号をランダムメモリマトリクス5800に対する情報のローディング、アンローディングを制御する

ためのタイミング信号として利用することもできる。

パラレルプロセッサはランダムメモリマトリクス5800中のデータに対して様々な処理を行うことができる。また、それらの処理の一部は同時に行うことが可能である。例えば、色情報は前述したようにデータの情報度に従い後で処理するか、あるいは分散させるために分離することができるし、受け取ったデータの情報内容を、前に第11図を用いて説明したように処理することもできる。

動作的な関係

コントローラ及びデータバスの数、及びこれらがメモリと共にどのような構成で用いられているかということは、MIMD及びSIMDに関してアーキテクチャを分類するのに役立つ。最も簡単な形式のものでは、「プロセッサ」は1つまたは2つ以上のコントローラと1つまたは2つ以上のデータバスで構成することができる。

第59図は、命令メモリ(5904, 5914, 5924, 5934)及びデータメモリ(5907, 5917, 5927, 5937)に接続された4つの別個の処理素子(5901, 5911, 5921, 5931)からなる典型的なMIMD構成を示す。命令メモリとデータメモリは別々に図示されているが、これらのメモリは実際は物理的に1つのメモリにまとめてもよい。各処理素子は、それぞれ2つの主ブロック、即ちコントローラ(5902, 5912, 5922, 5932)及びデータバス(5905, 5915, 5925, 5935)よりなる。命令メモリは命令バス(5903, 5913, 5923, 5933)を介して各コントローラへ制御命令を供給される。データメモリは、各コントローラの制御下にアクセスされ、データバス(5906, 5916, 5926, 5936)を介してデータバスへ接続されている。命令バスとデータバスは実際には同一バスとすることも可能であり、あるいはクロスバ構成になる一群のバスとした実施例も可能である。コントローラは一群の制御信号(5908, 5918, 5928, 5938)によってデータバスを制御する。

第59図のMIMD構成においては、各プロセッサは分散データまたは共用データに対して完全に独立の命令を実行することができる。

第60図は単一のコントローラ6002と命令メモリ6004を用いた一般的SIMD構成を示す。命令はバス6003を介してコントローラに供給される。この単一のコントローラは単一の命令群6000を生成し、この命令群は多重データバス(6010, 6020, 6030, 6040)をドライブする。各データバスはバス(6011, 6021, 6031, 6041)を介して各々のメモリ(6012, 6022, 6032, 6042)へ接続されている。簡単のために、各データバスのデータメモリへの接続の仕方は一通りしか示されていないが、実際は、クロスバ構成あるいは第8図に示すデータの逐次伝送のようにデータバスとデータメモリの接続の仕方は多様である。

第60図のSIMD構成では、多重データバスを単一の命令ストリームを用いて制御している。第60図に示すような一般的なSIMDの場合は、多重データバスに対してコント

ローラは1つしかない。第61図はMIMDモードで動作するよう構成したこの発明のマルチプロセッサシステムの一実施例を示す。図示実施例において、各パラレルプロセッサ(100, 101, 102, 103)は、クロスバスイッチ20を介して、メモリスペース10内のメモリを各々の命令メモリとして用いることができる。各パラレルプロセッサのコントローラ3002は各々異なる命令ストリームを受け取ることができる。MIMDオペレーションモードの構成となる場合、各パラレルプロセッサはバス40上の同期信号を無視する。各コントローラは制御信号3112で異なるデータバス3100を制御することができ、各データバスクロスバスイッチを介して異なるメモリにアクセスすることができるから、このシステムはMIMDモードで動作することができる。

第62図は第61図と同じハードウェアであるが、この場合はパラレルプロセッサはSIMDモードの構成になっている。このモードでは、既に第28図によって説明したように、単一の命令メモリがすべてのプロセッサに接続されている。SIMD構成をなす各パラレルプロセッサは同じ命令を受け、各コントローラは、一般に同じ制御信号を出す。例えば、制御信号はデータ依存性を考慮しなければならないために差異が生じる場合がある。バス40の同期信号は2つの目的のために用いられる：先ずこれらの同期信号はMIMDモードからSIMDモードのオペレーションに切り換わる時パラレルプロセッサをすべて同じ命令でスタートさせるのに用いられ、第2には、これらの同期信号は、いったんSIMDオペレーションでスタートした後、全部のパラレルプロセッサに等しく作用するとは限らないような何らかの事象のためにパラレルプロセッサが同期外れになるのを防ぐのに用いられる(例えば、2つのメモリが同じメモリにアクセスする場合、コンフリクト解消ロジックが一方のメモリを他方より先にメモリにアクセスさせるようになっている)。このようにして、第62図のシステムでは複数個のコントローラを用いているものの、システム全体としては第60図の従来のSIMD構成と同じ結果が得られる。前にも説明したように、MIMDモードで命令メモリとして使用されるメモリの一部をSIMDモードでは必要に応じてデータメモリとして使用する。

第63図は第61図及び第62図のハードウェアを用いて同期MIMDオペレーション用の構成とした実施例を示す。このモードの場合、各プロセッサは異なる命令を実行することができるが、それらの命令はバス40の同期信号により互いに同期関係に維持される。このオペレーションモードでは、一般に、プロセッサ間で異なる命令はほんの一部だけであり、プロセッサのメモリアccessを同じ相対的關係に保つことが大切である。

第64図は、第61図、第62図及び第63図に示すのと同じハードウェアを用いて可能な他の多くの構成中の一例を示す。図示例では、プロセッサ100と101は共通の命令メモリを共用すると共に、バス40の同期信号を用いること

によりSIMDオペレーションの構成になっている。これに対して、プロセッサ102と103は別個の命令メモリを使用しており、またバス40の同期信号を無視してすることによりMIMDモードで動作している。これ以外にもプロセッサをMIMD、SIMD、あるいは同期MIMDの各モードに割り当てるやり方は多数あり、そのような割り当て方による実施例が可能なこと、及びこれら3種類の各モードに任意の数のプロセッサを割り当てることができるということはもちろんである。

10 実施例の特徴

以下、この発明のマルチプロセッサシステムの実施例の重要な特徴を要約して説明する。

この発明のマルチプロセッサシステムの一実施例は、メモリソースから供給される命令セットにより各々動作して1つまたは2つ以上のアドレス指定可能メモリに対するデータの出入移動に依拠する多数の異なるプロセスを制御することが可能な n 個(n は正の整数)のプロセッサと、各々一意のアドレス指定可能空間を有する m 個(m は $m > n$ なる整数)のメモリソースと、上記メモリに接続され且つ上記プロセッサに接続されたスイッチマトリクスと、上記スイッチマトリクスをプロセッササイクル単位ベースで選択的に且つ同時にイネーブル化して上記のいずれかのプロセッサと上記いずれかのメモリとを相互接続し、上記メモリ空間からの命令セット及びそれ以外のアドレス指定可能メモリ空間からのデータをやりとりさせる回路群と、を具備したことを特徴とする。

このシステムにあつて、上記 n 個のプロセッサは、共用命令セットにより動作可能でこの共用命令セットからの同じまたは異なる命令ストリームにより並行処理能力において相互に動作することが可能な複数個のプロセッサと、及び異なる命令セットにより動作可能な少なくとも1つの他のプロセッサとを含み、これらのすべてのプロセッサを上記スイッチマトリクスを介して上記 m 個のメモリソースに接続することができるようにしたものであつてもよい。さらに、いずれかのメモリにアクセスする優先権を有するコンテンション回路群を組み込むことも可能であり、この回路群は何れか1つのメモリについて競合している時上記他のプロセッサに対して上記共用命令セットのプロセッサを優先させる回路群を含むことが望ましい。

これらのプロセッサは割込みを起こすことなく自律的に上記の各メモリに関して競合するものであつてもよく、コンテンション回路群はさらにシーケンシャルトークンバッシングを含むものであつてもよい。また、上記 n 個のプロセッサは、上記メモリへのまたは上記メモリからのデータ転送を自律的に制御するよう動作する転送プロセッサを含み、その自律的な転送がDSPプロセッサまたはRISCプロセッサからの終点命令の結果として起こるようにしたものであつてもよい。上記共用命令セット

プロセッサはDSPプロセッサとし、上記他のプロセッサはRISCプロセッサとすることが望ましい。

上記マルチプロセッサシステムは、さらに、上記m個のメモリをアクセスする回路群を含み、これらm個のメモリの中で上記n個のプロセッサによりアクセスされていない1つ以上のメモリを上記n個のプロセッサによるアクセスと干渉し合うことなく1つ以上の他のプロセッサによりアクセスすることができるようにしたものであってもよい。上記他のプロセッサは上記システムの内外により多くのデータを伝送するよう構成された転送プロセッサであってもよい。

また、上記マルチプロセッサシステムは、データ配線によるメモリ競合を防ぐための回路群、及びメモリ用途命令の配線を含むメモリ競合を防ぐための回路群を含むものであってもよい。

この発明のマルチプロセッサシステムの他の実施例は、サイクル単位ベースで実行される命令によって互いに独立に動作するよう構成された複数のプロセッサを有し、且つ複数のメモリ及び上記のいずれかのプロセッサといずれかのメモリとを相互接続する回路群を有し、且つ上記プロセッサの一群をそのすべてのプロセッサが同じ命令で動作するSIMDオペレーティングモードに接続構成する回路群、及びプロセッサにサイクル単位ベースで作用して少なくとも一部の上記プロセッサをSIMDオペレーティングモードのオペレーションから各プロセッサが別個の命令メモリより供給される別個の命令で動作するMIMDオペレーティングモードのオペレーションに切り換える回路群を含むことを特徴とする。

この発明の画像処理システムの一実施例は、メモリソースから供給される命令ストリームにより各々動作して各々一意のアドレス指定可能空間を有するm個のアドレス指定可能メモリに対するデータの出入移動に依拠する多数の異なるプロセスを制御することが可能なn個のプロセッサを有し（ただしmはnより大きな整数）、且つ上記メモリに接続され且つ上記プロセッサに接続されたスイッチマトリクスを有し、プロセッサが各々特定のプロセッサ-メモリ関係を有する複数の動作モードで動作することができるよう選択的に且つ同時にいずれかのプロセッサをいずれかのメモリに接続する回路群を含み、且つ上記の動作モード切り換えを行うためにいずれかのプロセッサからいずれか他のプロセッサに信号を伝送するためのプロセッサ間通信バスを含むことを特徴とする。

この発明のマルチマルチプロセッサシステムの他の実施例は、メモリソースから供給される命令ストリームにより各々動作して各々アドレス指定可能空間を有するm個のアドレス指定可能メモリに対するデータの出入移動に依拠する多数の異なるプロセスを制御することが可能なn個のプロセッサを有し、且つ上記メモリに接続されると共に上記プロセッサに接続されたリンクを有するス

イッチマトリクスを有し、且つこのスイッチマトリクスのリンクの少なくとも1つをスプリットして選択的に且つ同時に何れかのプロセッサをいずれかのメモリに接続し、上記メモリとこれに接続された上記プロセッサとの間で1つまたは2つ以上のアドレス指定可能メモリ空間からの命令ストリーム及びそれ以外のアドレス指定可能メモリ空間からのデータをやりとりさせるようにしたことを特徴とする。

この発明の処理装置の実施例は、各々独自の命令を実行し得る複数のプロセッサを有し、上記の各プロセッサに関連させて設けられ、プロセッサに他のどのプロセッサを同期させるかを決定する制御回路群及び上記各プロセッサに関連させて設けられ、互いに同期した他のプロセッサと同期させるべき命令の境界を決定すると共にそれらの境界の間にフラグをセットする命令応動回路群を伴い、且つ実行待機モードを確立する各プロセッサに設けられた回路群を含み、プロセッサに各フラグがセットされている間は各プロセッサにより各々と同期させることが決定された他のプロセッサがすべて実行待機モードになるまでそのプロセッサの中の如何なる命令の実行をも禁止するよう動作するロジックを有することを特徴とするものである。

この発明のマルチプロセッサシステムの他の実施例は、各々一意のアドレス指定空間を有するm個のメモリを具備し、上記m個のメモリ全アドレス指定可能空間はnビットの単一アドレスワードによって決定され、且つ上記アドレスワードのビット値に従い上記m個のメモリのアドレス指定可能記憶場所へのアクセスを制御するメモリアドレス生成回路を具備し、且つ現アドレスワードに加えるべきインデックス値をアクセプトして次のアドレス記憶場所を指定するためにビット間のキャリーオーバー信号を有する加算回路群を具備し、且つ通常は一つ隣のメモリアドレスワードをトグルさせるはずの上記ワードの一部のビットからキャリーオーバー信号をダイバートさせ、このキャリーオーバー信号にメモリアドレスワードのリモートビットをトグルさせるよう動作する回路群を具備したことを特徴とする。

この発明の2進ストリング中の「1」の数を示す回路の一実施例は、第1及び第2の入力と出力を有するANDゲートと、第1及び第2の入力と出力を有するXORゲートとを具備し、このXORゲートの第1の入力は上記ANDゲートの第1の入力に接続されており、XORゲートの第2の入力はANDゲートの第2の入力に接続されており、上記ANDゲート及びXORゲートの第2の入力は2進ストリングの1ビットを供給されると共に、XORゲートの出力が2進ストリングのビット中の「1」の数を表す2進数を出力するようにしたことを特徴とする。

この発明のマルチプロセッサシステムの他の実施例は、1つまたは2つ以上のアドレス指定可能なメモリからのデータの移動に依拠する異なるいくつかのプロセス

を制御するためにメモリソースから供給される命令ストリームにより動作可能な n 個のプロセッサを具備し、且つ各々一意にアドレス指定可能な m 個のメモリソースを具備しており、これらのメモリの一部は上記プロセッサ用の命令ストリームを共用記憶するよう構成され、その他の上記メモリはプロセッサ用にデータを記憶するよう構成されており、且つ上記プロセッサとメモリとの間に通信リンクを設定するためのスイッチマトリクスを具備し、このスイッチマトリクスは特定のプロセッサと上記メモリの中の命令ストリームを記憶する特定のメモリとの間で専用の通信を行うためのある種のリンクが具備されており、且つスイッチマトリクスリンクを再構成してそれまでは命令用に用いられたメモリへのデータアクセスを可能にする回路群、及びこの回路群と同時に動作してすべてのプロセッサをある種のリンクの中の特定の1つに接続し、これによって上記リンクに関連する命令メモリからの命令をすべてのシステムプロセッサに伝送させるよう構成された回路群を具備したことを特徴とするものである。

この発明の画像システムの一実施例は、各々ピクセルに関連する複数のデータビットで構成し得る複数のピクセルよりなるイメージのイメージ入力を有すると共に、メモリ、各入力イメージのピクセルをメモリへ伝送するイメージバス、及び供給されたイメージをメモリに記憶されたパラメータに従い解釈するためにこれらのパラメータを上記に供給された各イメージのピクセルに適用する回路群を具備したことを特徴とするものである。

この発明のスイッチマトリクスの一実施例は、複数の第1のポートと複数の第2のポートとを相互接続するマルチプロセッサシステム用のスイッチマトリクスであって、各々上記第1のポートの特定の1つに付随させて設けられた複数の縦方向バスと、個々に動作可能な複数の又点と、上記第2のポートに接続されていて、上記又点の中のイネーブル化された又点を介して上記第1のポートの1つを上記第2のポートのいずれか1つに接続する複数の横方向バスとを有し、且つ上記第2のポートの中の競合するポート間の競合処理を行って上記各縦方向バスに接続するためのこれら各縦方向バスに付随させて上記各又点に設けられた回路群を含むことを特徴とするものである。

このスイッチマトリクスにあって、特定縦方向バスの各又点の競合処理回路は、上記縦方向バスに関連する上記第1ポートの永久アドレス名を受け取る回路群を含むものであってよく、かつ特定横方向バスの各又点の競合処理回路は、上記横方向バスに関連する上記第2ポートより上記マトリクスを介して特定の第1ポートへ到る所望の接続のアドレス名を適時受け取る回路群を含むものであってよい。

上記各又点の競合処理回路は、2つ以上の上記第2ポートから同じ第2ポートのアドレス名を受け取った場合

に上記各縦方向バスに関連する上記又点の1つに第1の優先順位を与えるトークンバスロジックを含むものであってよく、また上記第1ポートが複数のメモリで、上記第2ポートが複数のプロセッサであってもよい。上記スイッチマトリクスはシングルチップ上に、望ましくは上記又点と共に構築することができ、また上記競合処理回路は上記第1及び第2ポートと共に上記シングルチップ上に空間的に分散配設する。

要約

- 10 以上、この発明を特定の実施例に基づき説明してきたが、当業者であれば種々の変更、修正を示唆することが可能であり、この発明はそのような変更、修正も発明の範囲内に含まれるとみなすものである。また、本願は画像処理を中心として開示、説明したが、この発明のシステムがグラフィックス（図形処理）、信号処理、スピーチ処理、ソナー、レーダー及びその他の高密度リアルタイム処理用にも同様に使用可能であるということは明らかである。

【図面の簡単な説明】

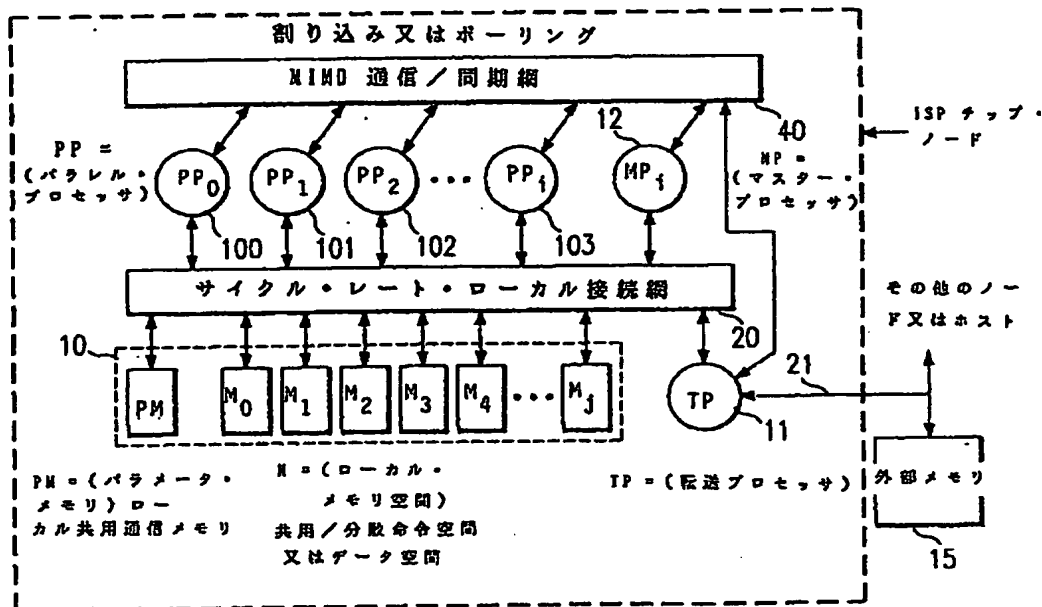
- 20 第1図及び第2図はこの発明の画像処理システムの構成を示すブロック図、第3図は一連の画像処理システムを相互接続して拡張システムとした例のブロック図、第4図はこの発明のシステムにおけるパラレルプロセッサとメモリとを相互接続するクロスバススイッチマトリクスの詳細図、第5図及び第6図は従来技術によるシステムの例を示すブロック図、第7図はその改良された構成例を示すブロック図、第8図及び第9図は従来技術におけるプロセッサとメモリの相互作用の概念を示すブロック図、第10図はこの発明による構成可変型マルチプロセッサの一例の構成を示すブロック図、第11図は画像処理システムにおけるアルゴリズム及び制御の流れを示す機能ブロック図、第12図乃至第15図はSIMD及びMIMDオペレーションモードのイメージピクセルの流れを示すブロック図、第16図はプロセッサ間の割込みポーリング通信を示す模式図、第17図はクロスバススイッチにより相互接続されたプロセッサとメモリのレイアウト模式図、第18図及び第19図クロスバススイッチの又点の詳細図、第20図はメモリアccessのコンテンションロジックのタイミング図、第21図乃至第23図はプロセッサ間の同期制御の説明図、第24図乃至第27図はスライス式アドレス指定技法の説明図、第28図はSIMD/MIMDオペレーションモードにおける命令データメモリの構成切り換えの詳細を示す説明図、第29図はマスタープロセッサの詳細ブロック図、第30図乃至第34図はパラレルプロセッサの詳細ブロック図、第35図乃至第45図はパラレルプロセッサのオペレーション方法の説明図、第46図乃至第48図はパーソナルコンピュータとして用いられるイメージプロセッサの説明図、第49乃至52図はリモート及びローカルベースによる画像システムの使用形態を示すブロック図、第53図は画像システムの機能ブロック図、第54図は「1」計数回路

マトリクス論理回路図、第55図は最小規模化した「1」計数回路の論理図、第56図は「1」計数回路の応用形態の一例の説明図、第57図は転送プロセッサのブロック図、第58図はVRAMと共に使用する状態を示すパラレルプロセッサのブロック図、第59図乃至第64図は色々なオペレーションモード間の関係を示す説明図である。

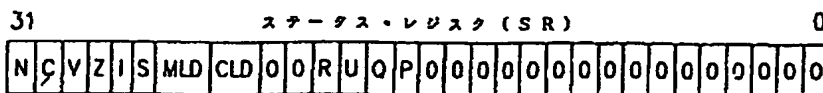
10……メモリ、
11……転送メモリ、
12……マスタープロセッサ、
13……データキャッシュ、
14……命令キャッシュ、
15……外部メモリ、
20……クロスバスイッチマトリクス、
40……通信バス、
100～103……パラレルプロセッサ、
170……フレームコントローラ、
401～403……スリーステートバッファ、
2101～2104……レジスタ及びゲート、
2504～2506……加算器セル、
2507, 2508……マルチプレクサ、
4900……イメージシステムプロセッサ及びメモリ、
4901……モデム、

* 4902……制御コンソール、
4903……ディスプレイ、
4904……データ収集装置、
4905……コントローラエンジン、
4906……CCDユニット、
4907……光学系、
4909……プリントエレメント、
5000……イメージシステムプロセッサ、
5001……光りディスク、
10 5002……ハードディスクドライブ、
5003……VRAM、
5004……CCDユニット、
5007……統計学的記録維持、
5008……ラッチ、
5009……制御対象機構、
5010……出力ディスプレイ、
5101……イメージシステムプロセッサ、
5102……メモリ、
5103……カメラ、
20 5104……フラットパネルディスプレイ、
5105……CCD、
* 5424……マトリクス。

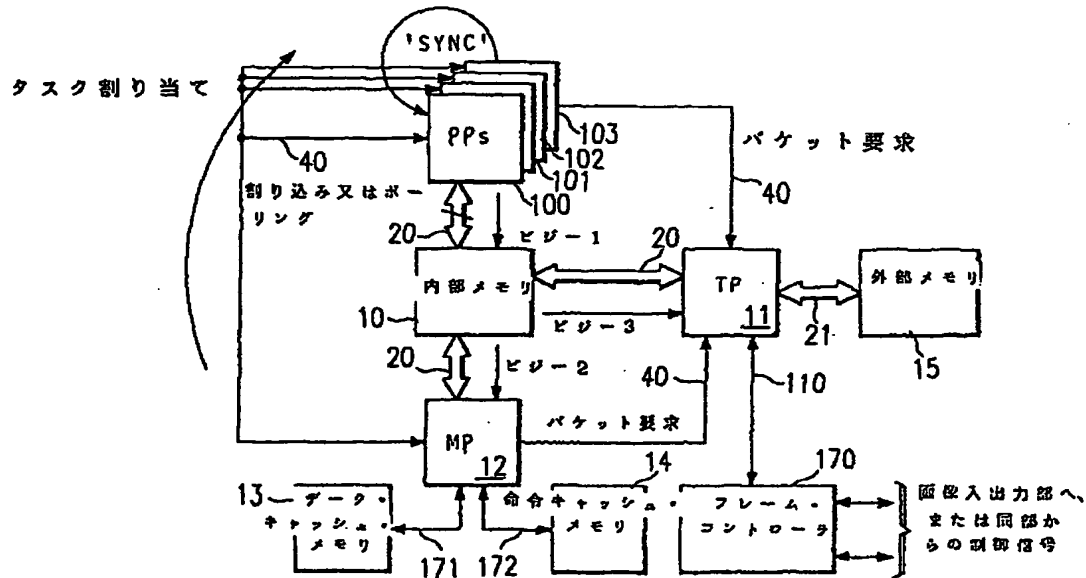
【第1図】



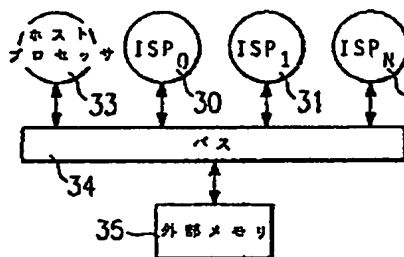
【第34図】



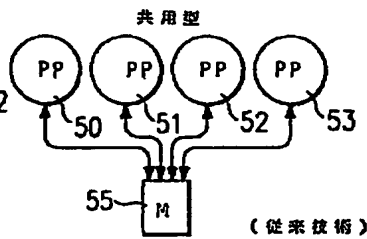
【第2図】



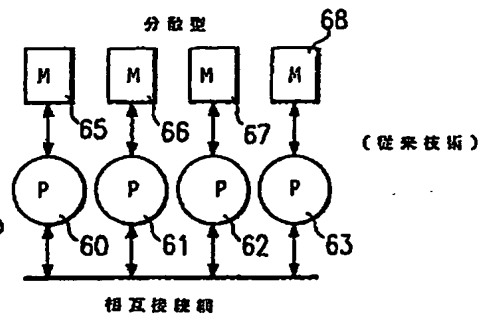
【第3図】



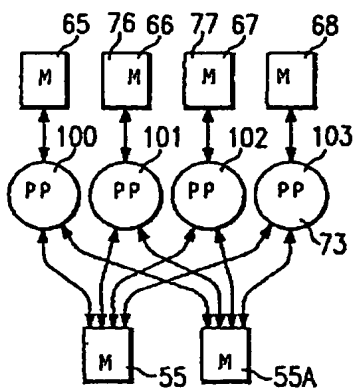
【第5図】



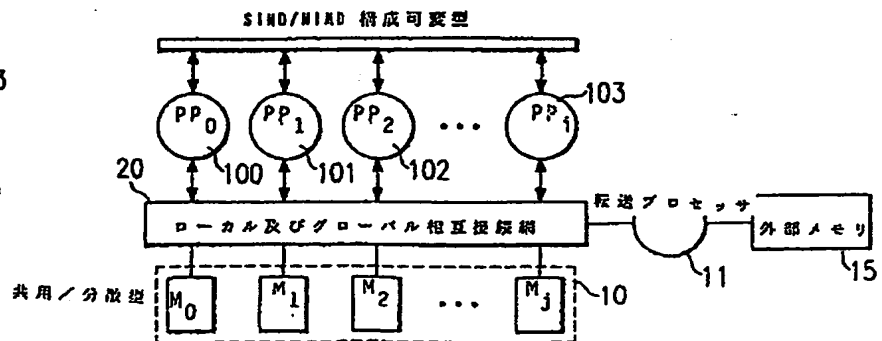
【第6図】



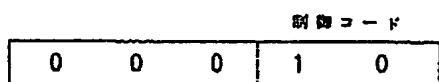
【第7図】



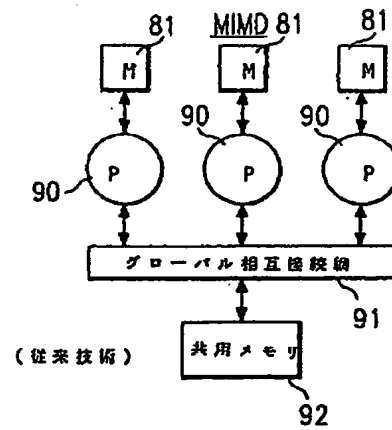
【第10図】



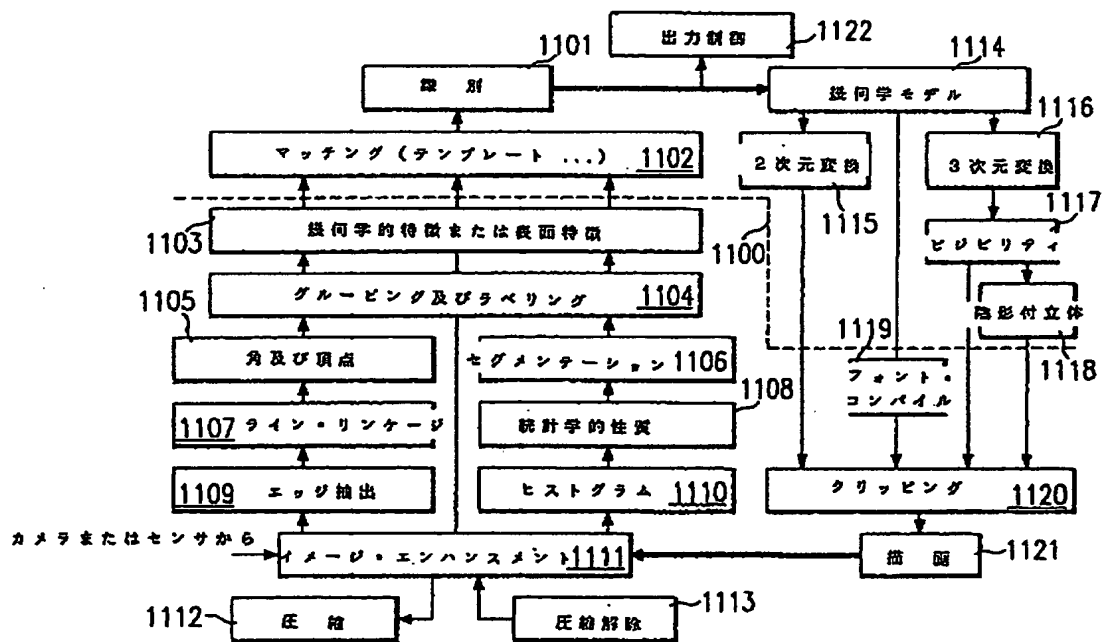
【第47図】



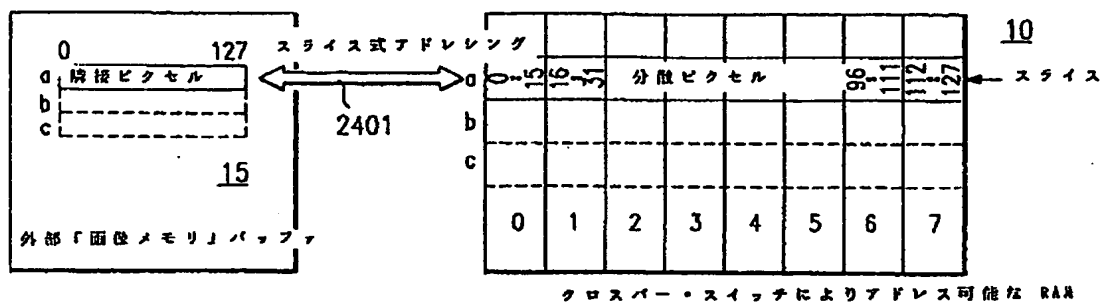
【第9図】



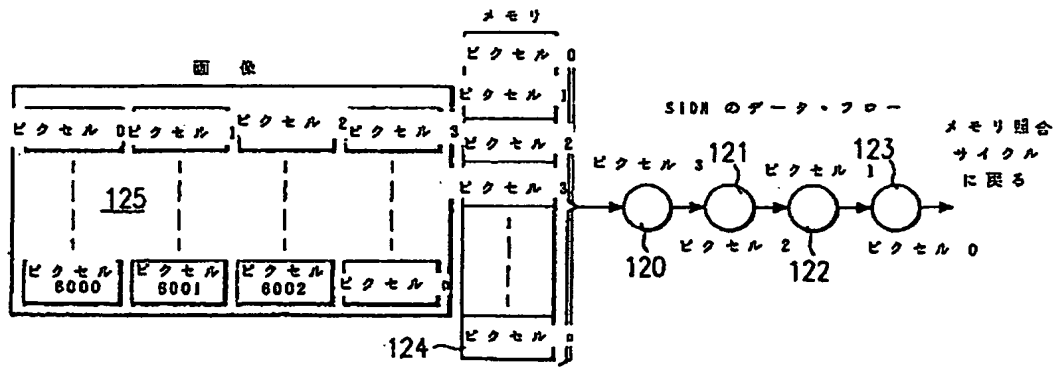
【第11図】



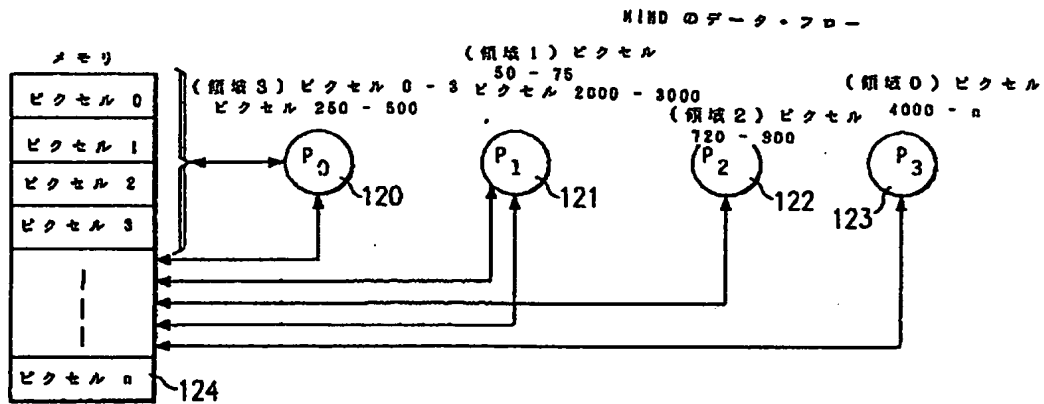
【第24図】



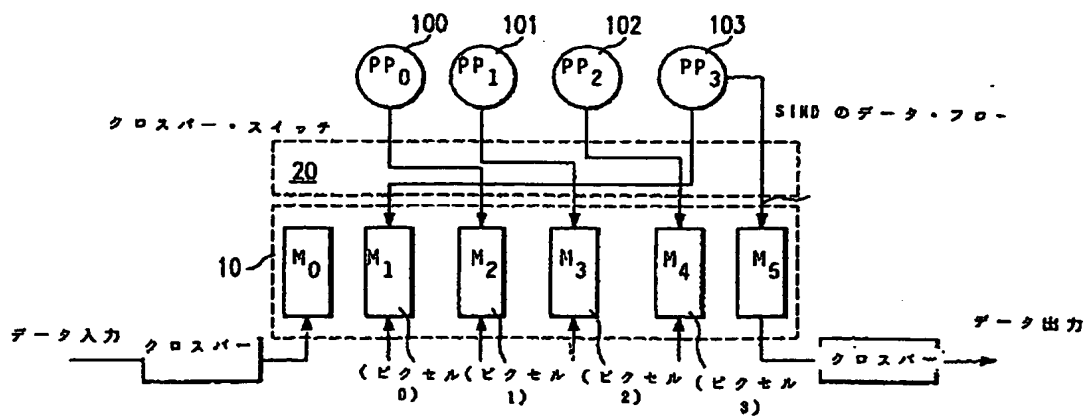
【第12図】



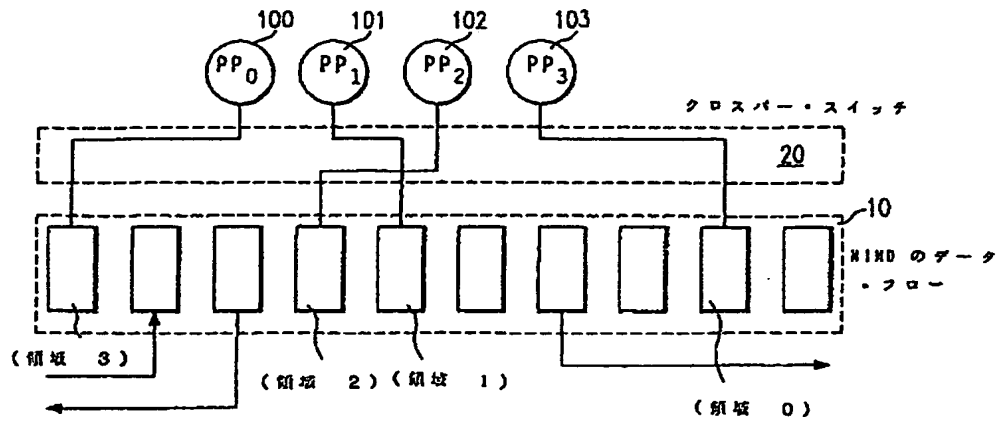
【第13図】



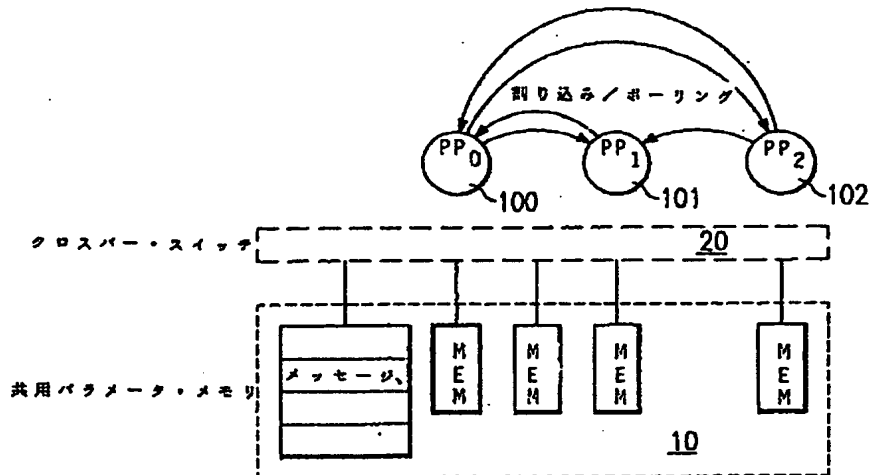
【第14図】



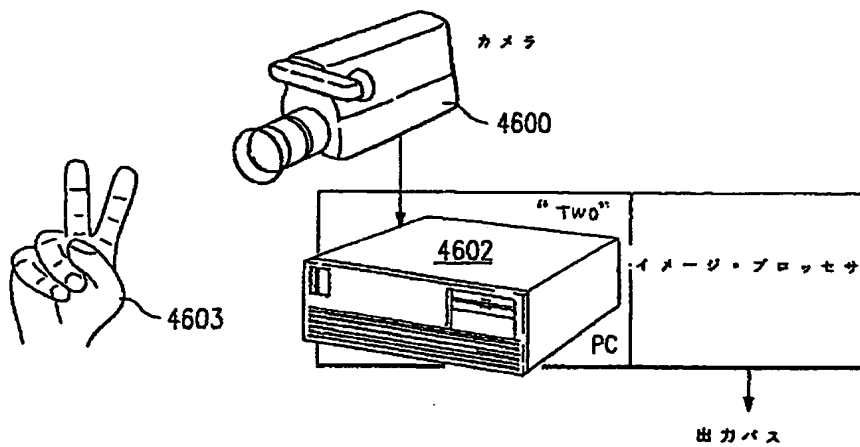
【第15図】



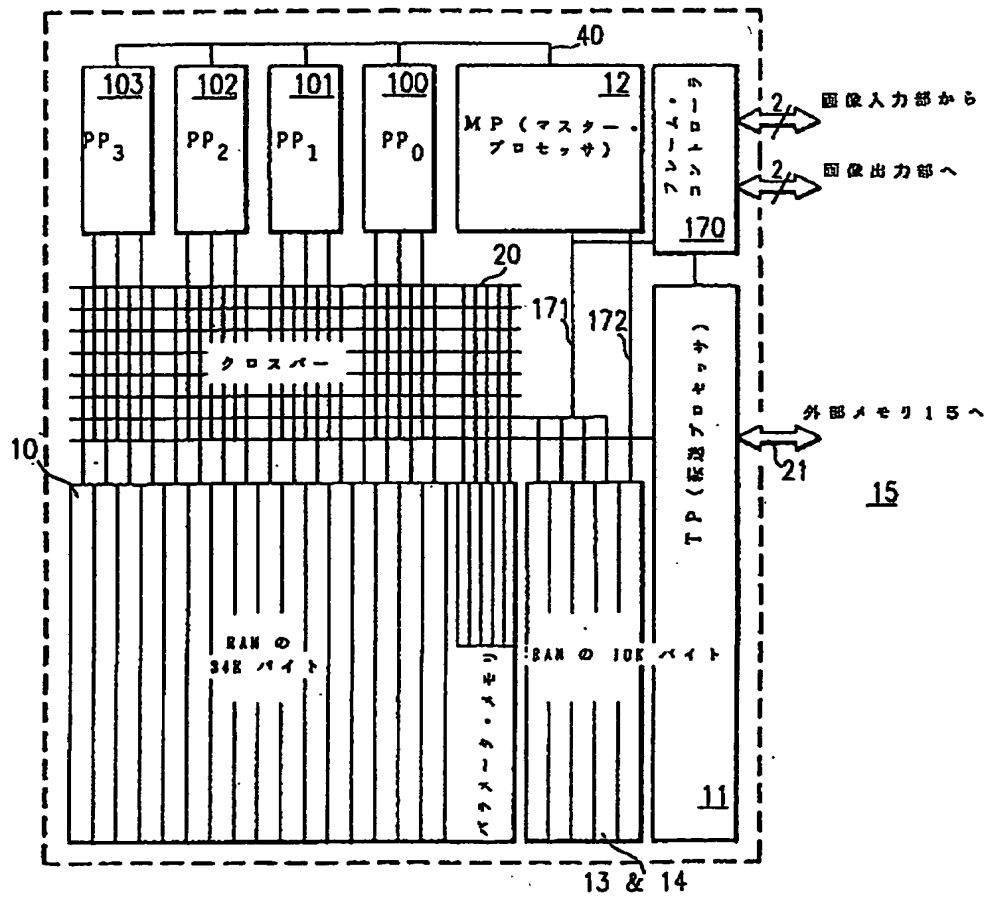
【第16図】



【第46図】



【第17図】



【第42図】

ロード: (符号拡張なしの場合)

バイト番号

ソース・データ

3 2 1 0

(メモリ)

0000h = D C B A

0004h = H G F E

(レジスタ)

先行 = ? ? ? ?

16 - ビット

32 - ビット

OP.	ADD.	ロード...	REG 値	OP.	ADD.	ロード...	REG 値
LD	0000h	0 0 B A	0 0 B A	LD	0000h	D C B A	D C B A
LDU	0002h	- - - -	0 0 B A	LDU	0004h	- - - -	D C B A
LD	0001h	- - - B	? ? ? B	LD	0001h	- D C B	? D C B
LDU	0003h	0 0 C -	0 0 C B	LDU	0005h	E - - -	E D C B
LD	0002h	0 0 D C	0 0 D C	LD	0002h	- - D C	? ? D C
LDU	0004h	- - - -	0 0 D C	LDU	0006h	F E - -	F E D C
LD	0003h	- - - D	? ? ? D	LD	0003h	- - - D	? ? ? D
LDU	0005h	0 0 E -	0 0 E D	LDU	0007h	G F E -	G F E D

研究化學 20

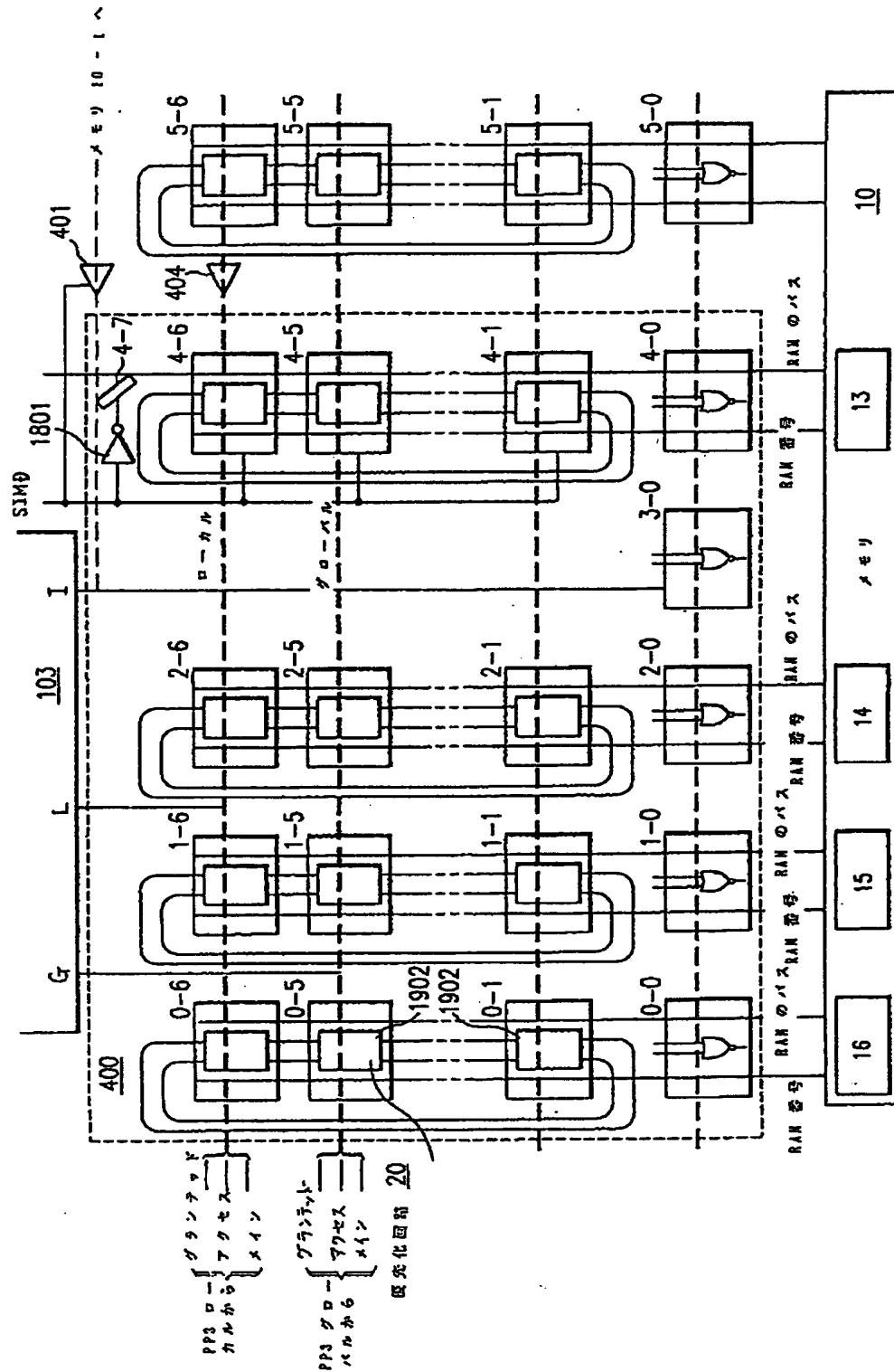


図 10

セッサ / プロセッサから RAN 番号

ランチャド番号
グローバル・アドレス番号
グローバル・アドレス /
データ・バス

次の交叉点へ

1934 1933 1932

1905

1958

1957

交叉スイッチ

TP 交叉へ

1950 1941

1943

1949

1952

1906

1942

1915

1940

1903

1911

1920

1921

1910

1923

1924

1901

1902

1930

1912

1913

1914

1944

1956

1945

1954

1951

1904

1946

1955

1947

1907

1948

1931

プロセッサへ

RAN のバス

1-5

イネーブル SIND

図 10 記号説明

トランザクタ

クロック

AND

OR

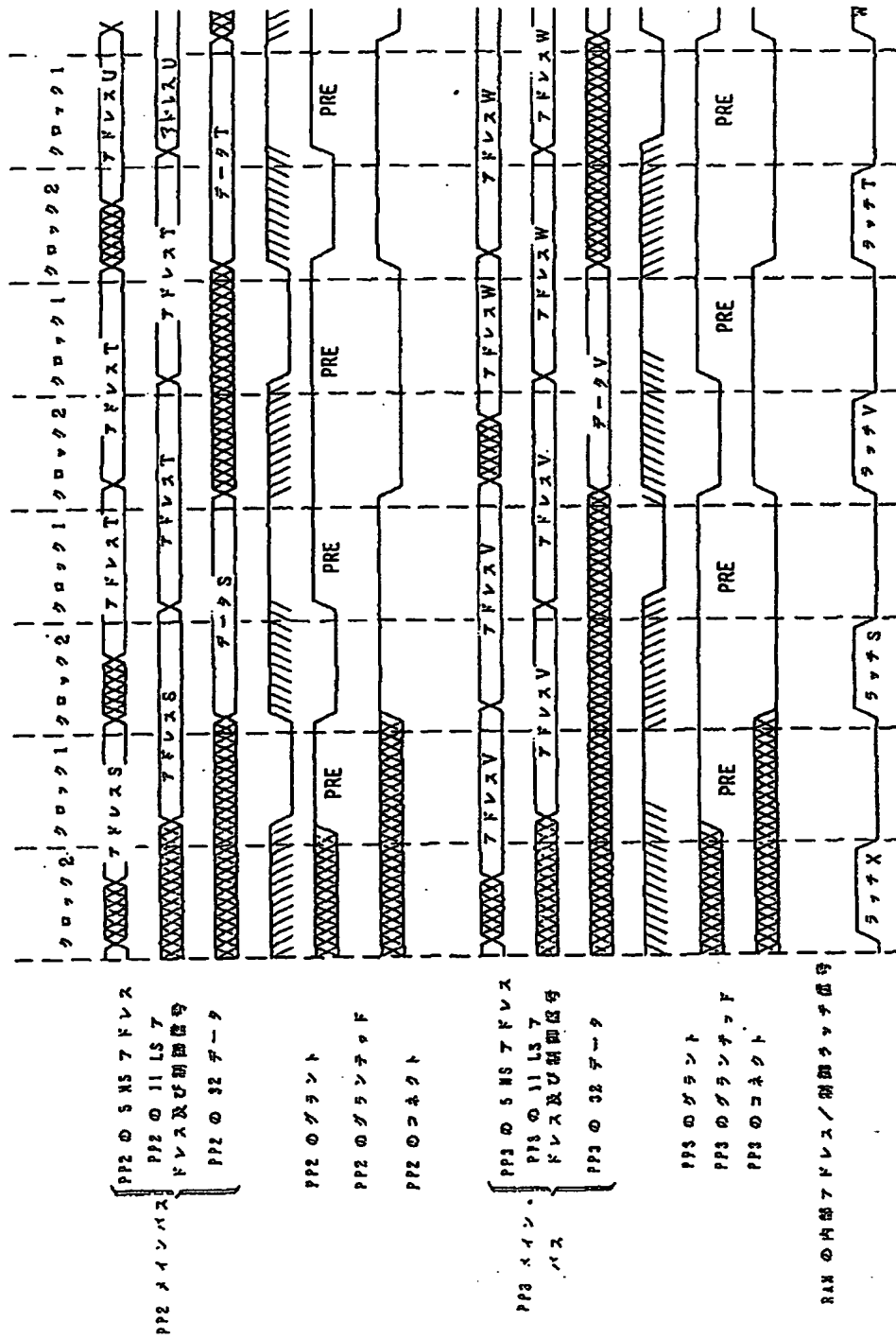
NOT

1-5

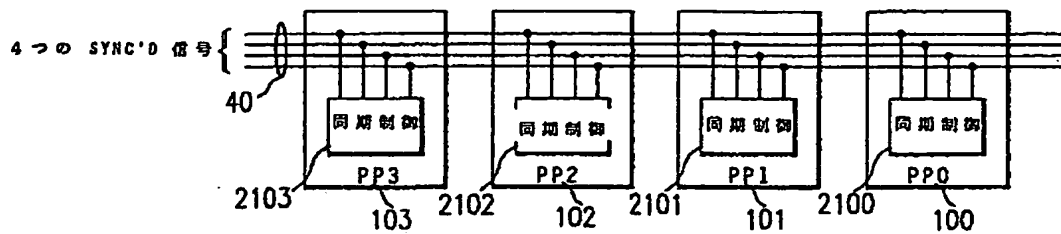
全ての下部交叉点を介してメセリへ

全ての下側隙又歯を介してAセリへ

【第20図】



【第21図】



【第27図】

>0000	>0800	>1000	>1800	>2000	>2800	>3000	>3800
64 PIXELS	64 PIXELS	64 PIXELS	>1900 >193F	>2100 >213F	64 PIXELS	64 PIXELS	64 PIXELS
			>1940				
>07FF	>0FFF	>17FF	>1FFF	>27FF	>2FFF	>37FF	>3FFF
0	1	2	3	4	5	6	7

通常演算

$$\begin{aligned}
 &\left\{ \begin{array}{l} >193F - 0001100100111111 \\ + \\ >0001 - 0000000000000001 \\ \hline >1940 - 0001100101000000 \end{array} \right.
 \end{aligned}$$

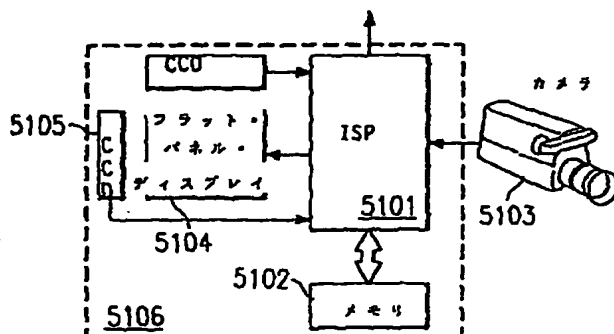
スライス番号

スライスマスク

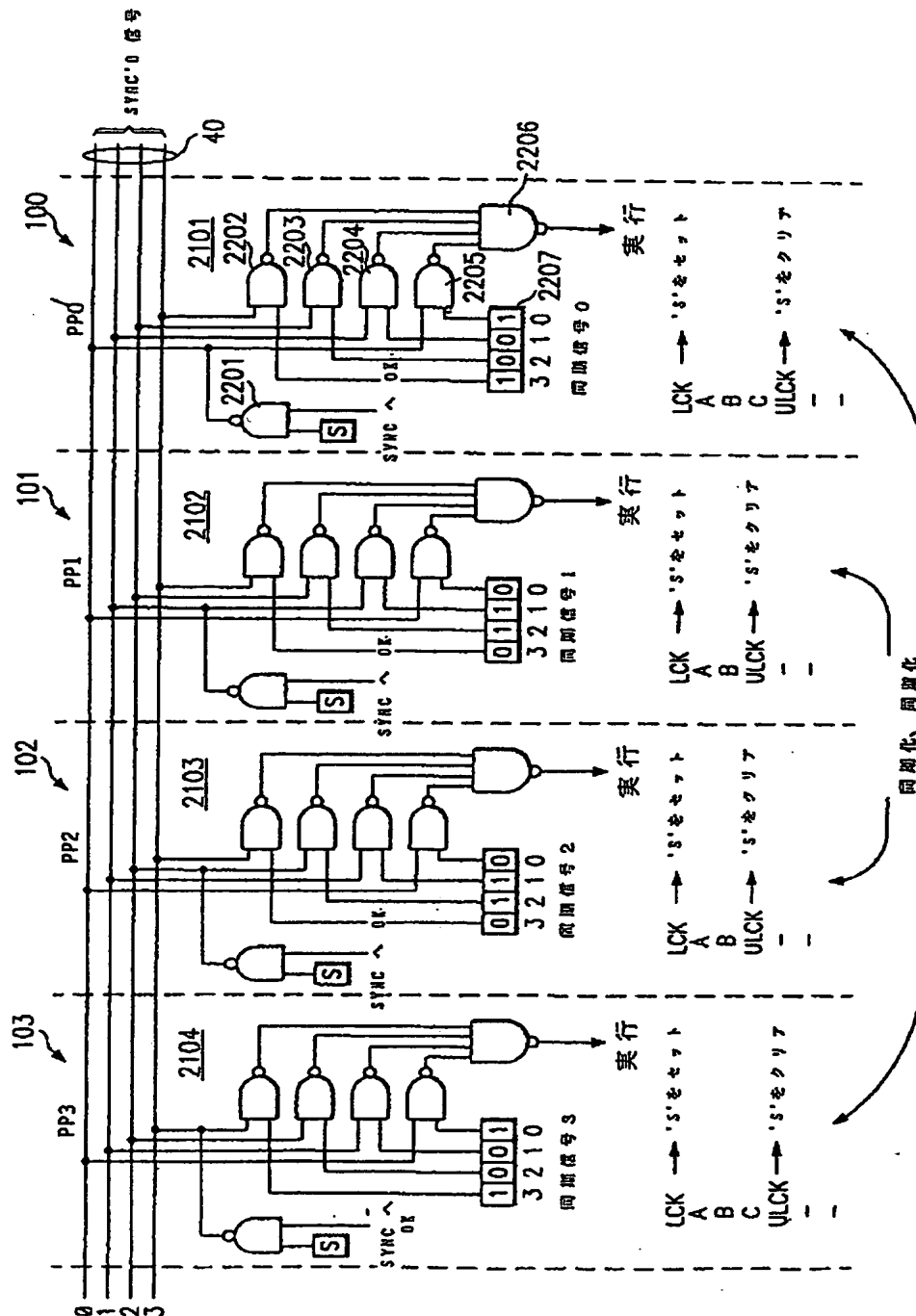
$$\begin{aligned}
 &\left\{ \begin{array}{l} >193F - 0001100100111111 \\ (スライスマスク) \quad 0000011111000000 \\ \hline + \quad \text{キャリー} \\ >0001 - 0000000000000001 \\ \hline >2100 - 0010000100000000 \end{array} \right.
 \end{aligned}$$

【第51図】

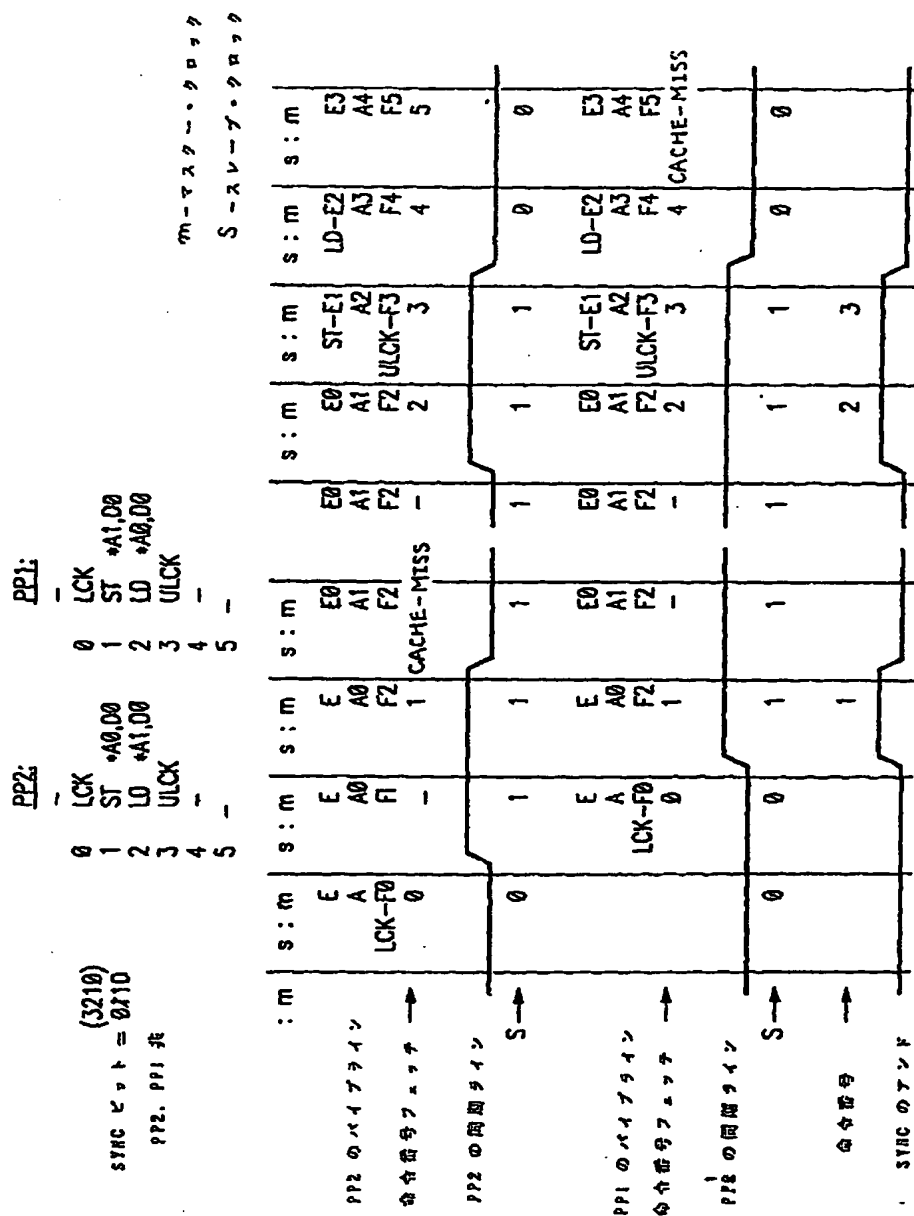
ホスト・プリンタ・ポート



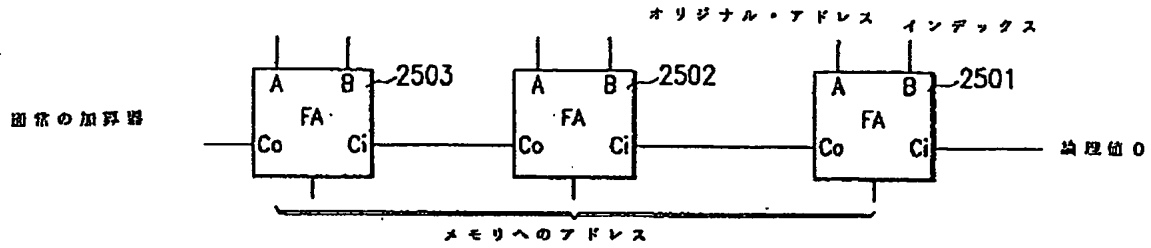
【第22図】



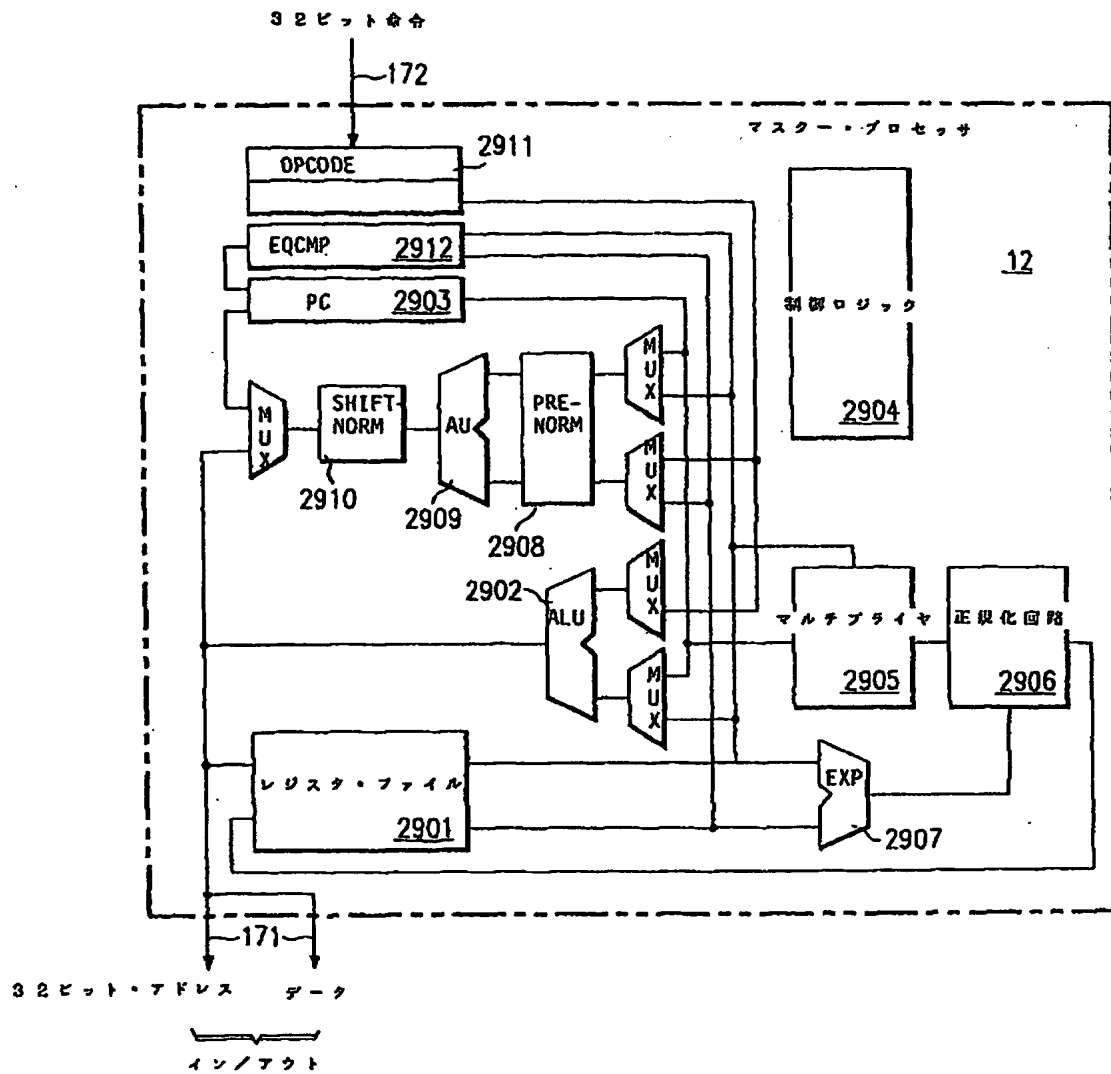
【第23図】



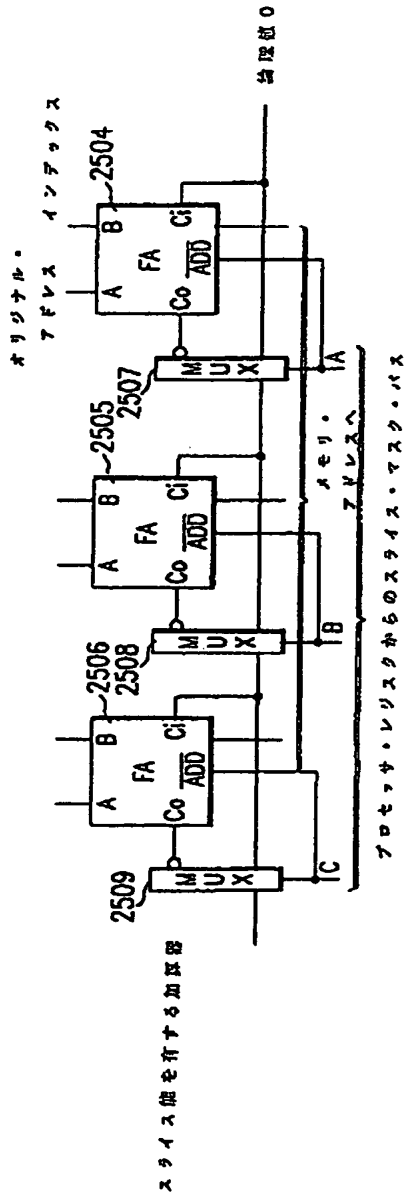
【第25図】



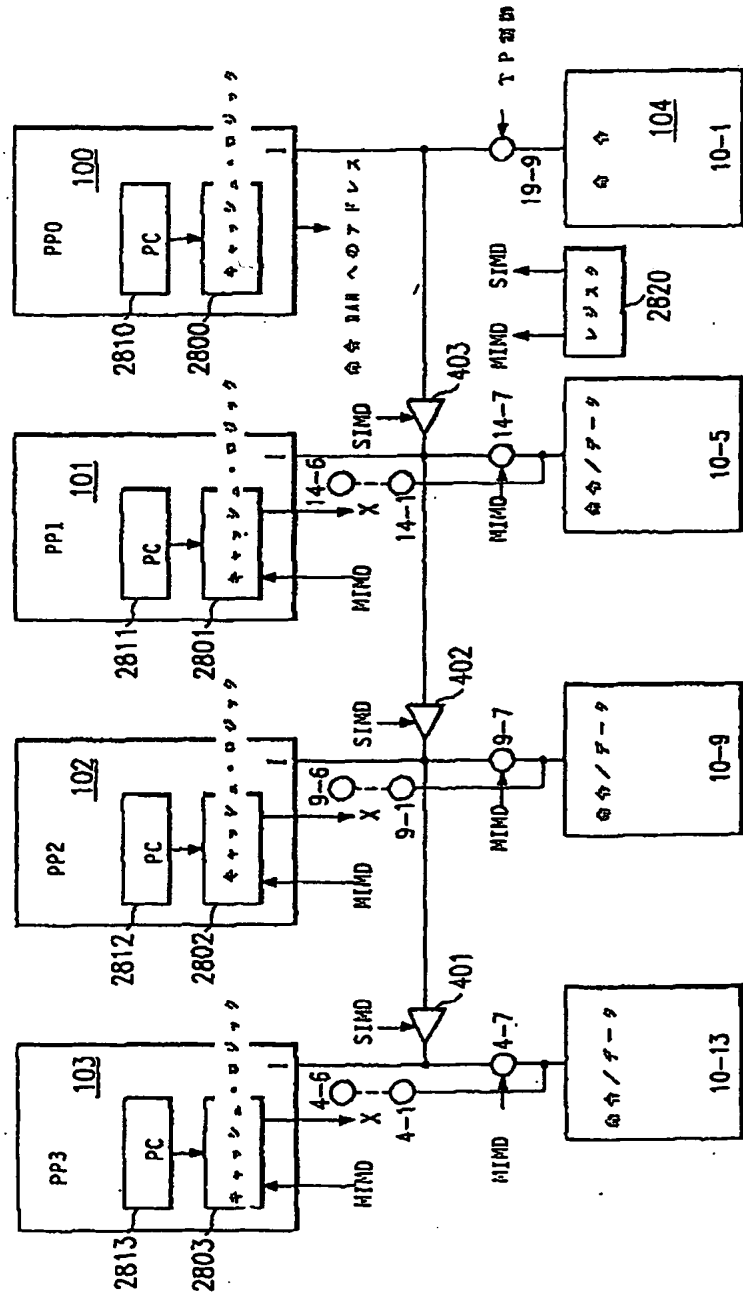
【第29図】



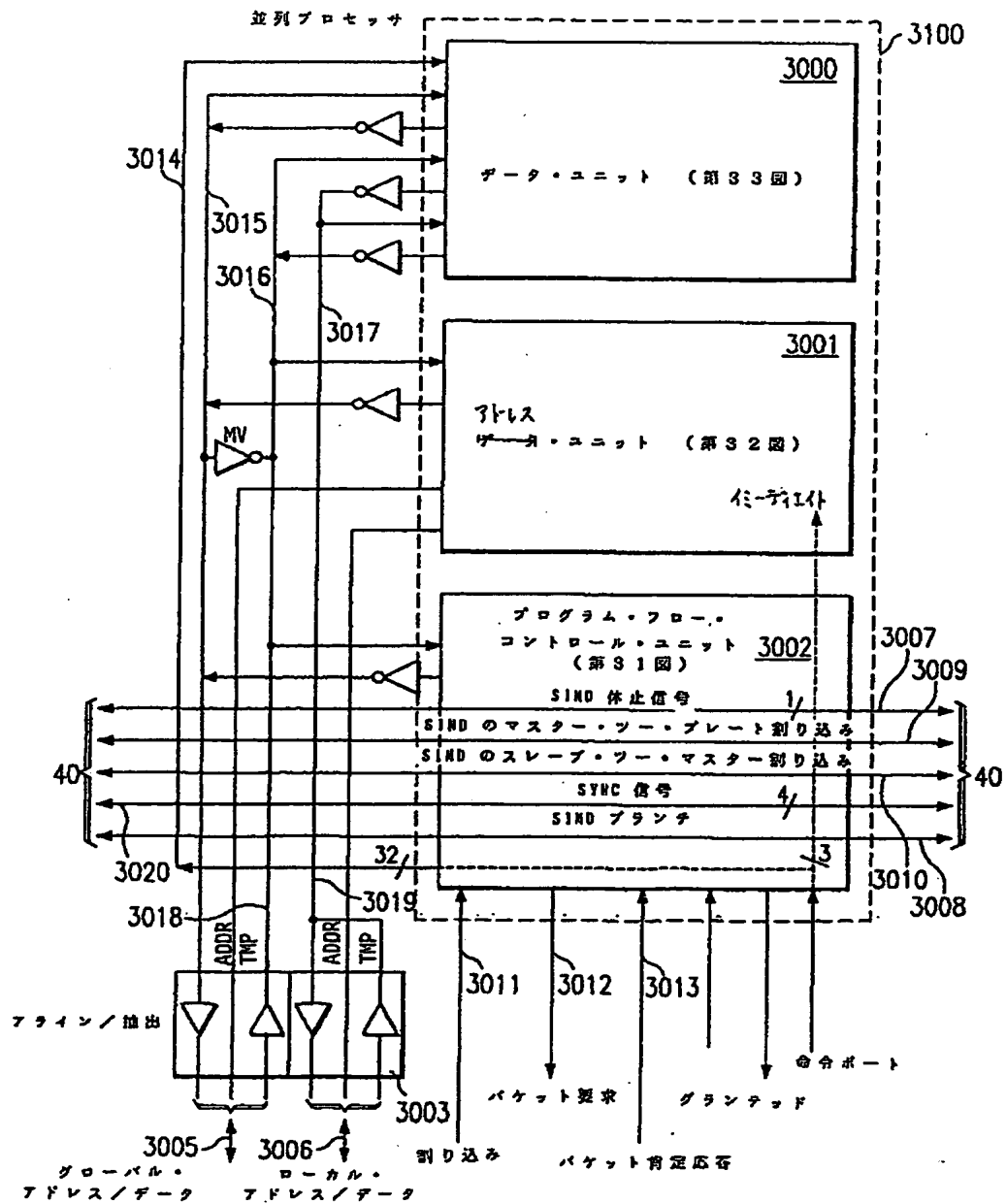
【第26図】



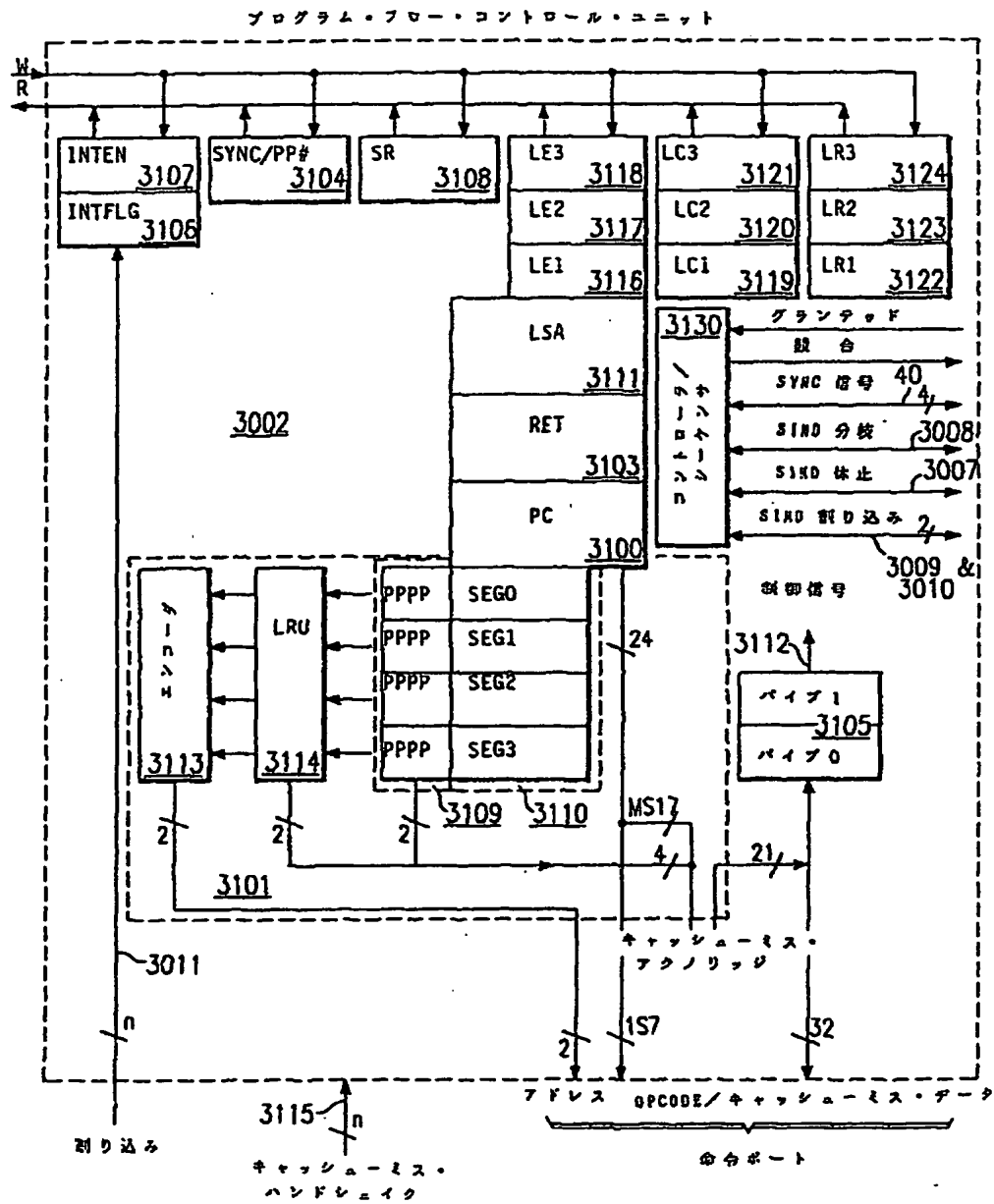
【第28図】



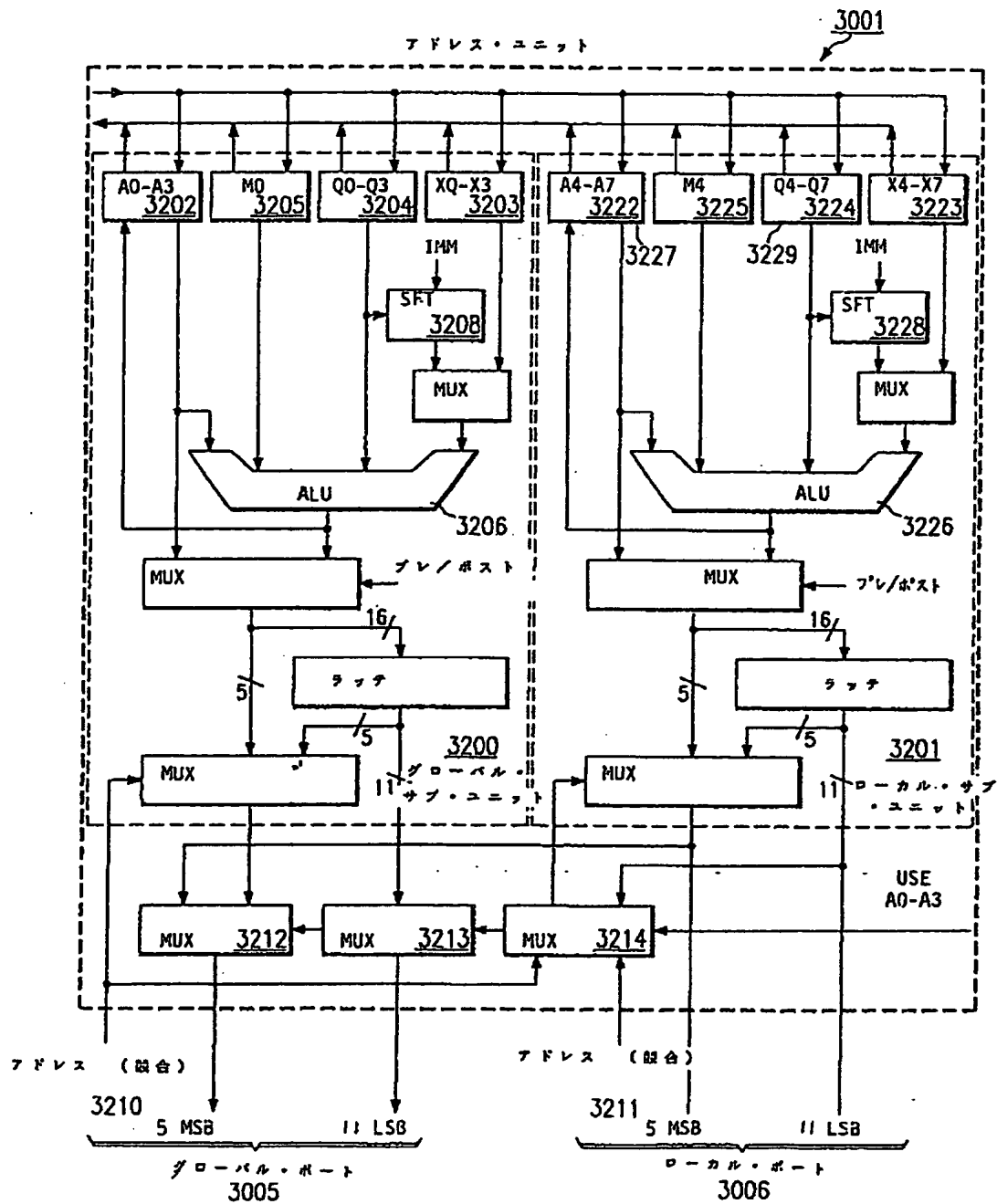
【第30図】



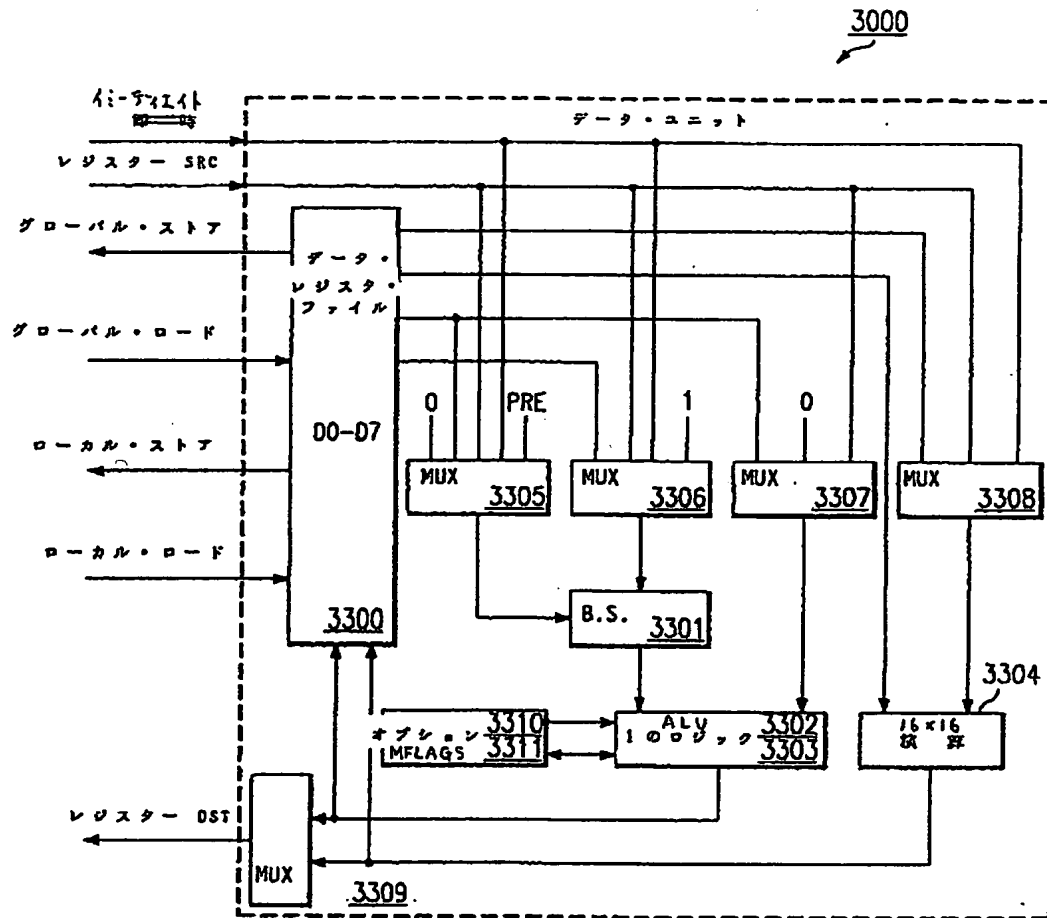
【第31図】



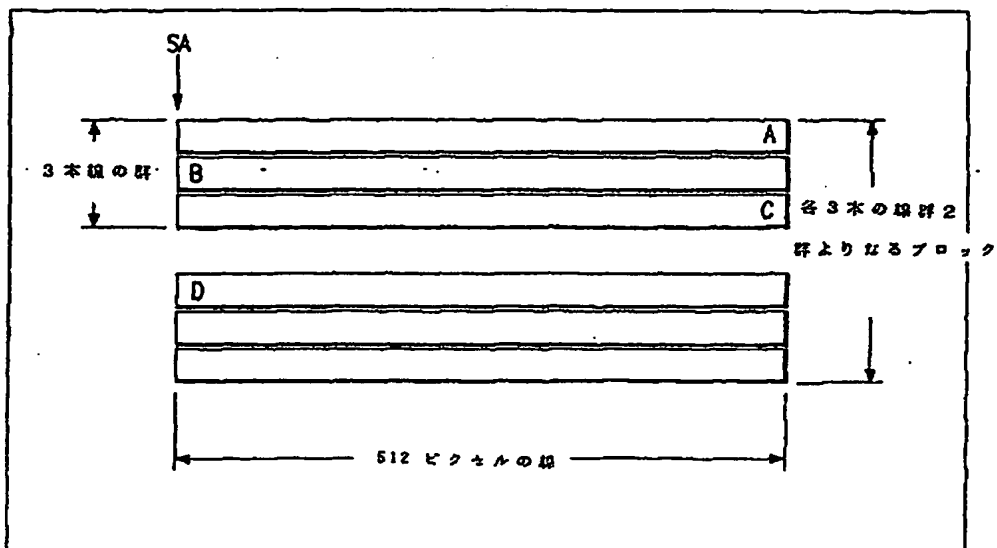
【第32図】



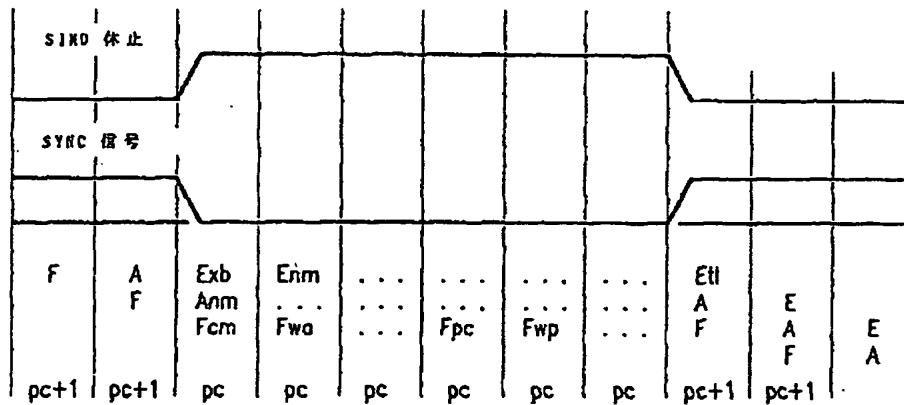
【第 3 3 図】



【第 4 5 図】

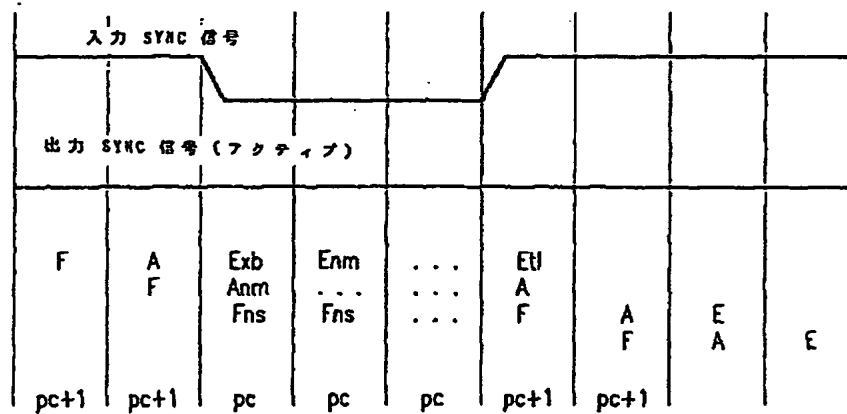


【第35図】



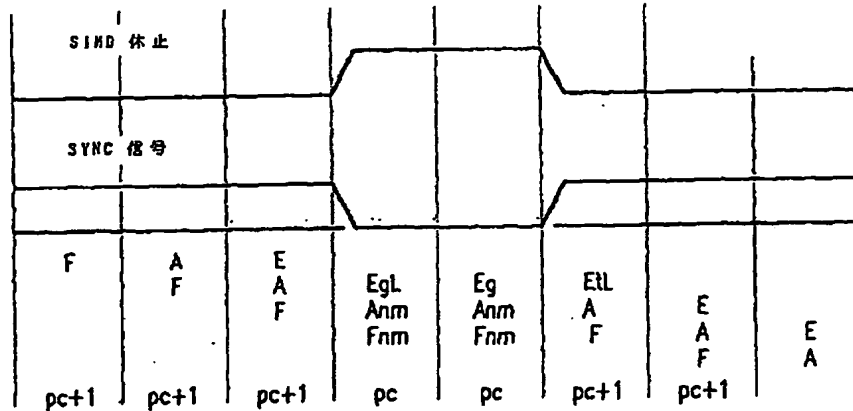
Fns - キャッシュミス
 Ans - アドレス・ユニットのマスター・フューズがなく、レジスタ修正なし
 Exb - クロスバー・スイッチ・アクセス発生。メモリに完全に接続する。テンポラリー・ラッチに完全にロードする。データ・ユニット・オペレーションのマスター・フューズがキルされている。
 Enm - データ・ユニットのマスター・フューズなし
 Fpc - 17 からの キャッシュ・ミス・アド/ラジ付与
 Fwp - 17 からの PC 及びセグメント番号返還
 Fsp - セグメントのプレセント・アドレスを持ってセット
 Et1 - 先行レジスタにテンポラリー・ラッチのデータを完全に (ロードする)。データ・ユニットは ALU/XPY オペレーションを実行。

【第41図】



Fns - 同相状態なし。パイプがロードされていない。
 Fc 修正なし。
 Ans - アドレス・ユニットにマスター・フューズなし。アドレス・レジスタが修正されていない。
 Exb - クロスバー・スイッチ・アクセス発生。完全にメモリに接続し。テンポラリー・ラッチに完全にロードする。データ・ユニット・オペレーションのマスター・フューズがキルされている。
 Enm - データ・ユニットのマスター・フューズなし。
 Et1 - 先行レジスタにテンポラリー・ラッチのデータを完全に (ロードする)。
 データ・ユニットは ALU/XPY オペレーションを実行

【第36図】



161 - グローバル・バス、ローカル・バス両方に接続。
データ・バスにマスタ・フェーズなし。
162 - フェーズ上にマスタ・フェーズなし。
パイプラインされていない。
163 - グローバル・バス上に接続。ローカル・バスに接続。
メモリに接続するか、サンプリング・レジスタにロ
ードするか、データ・バスにマスタ・フェーズなし。
164 - 実行レジスタにアドレスをロードする。ロードのデータを強
制に（ロードする）。データ・バスは ALU/MPV オペレ
ーションを待機。
165 - アドレス・バスにマスタ・フェーズなし。
アドレス・レジスタが接続されていない。

【第43図】

記 憶

ソース・データ = 0 C B A (レジスタ)

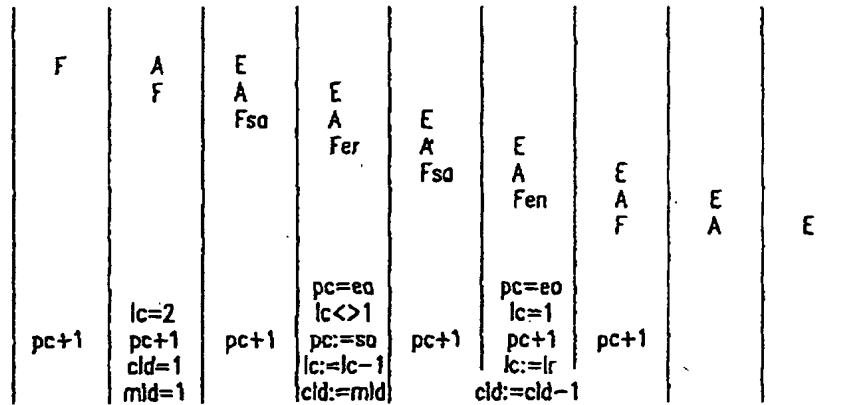
先行データ = バイト番号
3 2 1 0
- - - - (メモリ)

0000h = ? ? ? ?
0004h = ? ? ? ?

32 - ビット

OP.	ADD.	16 - ビット 記憶 ...	0000h 0004h	OP.	ADD.	記憶 ...	REG 値
ST	0000h	0 0 B A	? ? B A ? ? ? ?	ST	0000h	0 C B A	0 C B A ? ? ? ?
STU	0002h	- - - -	? ? B A ? ? ? ?	STU	0004h	- - - -	0 C B A ? ? ? ?
ST	0001h	- - A -	? ? A ? ? ? ? ?	ST	0001h	C B A -	C B A ? ? ? ? ?
STU	0003h	- B - -	? B A ? ? ? ? ?	STU	0005h	- - - D	C B A ? ? ? ? D
ST	0002h	- A - -	? A ? ? ? ? ? ?	ST	0002h	B A - -	B A ? ? ? ? ? ?
STU	0004h	B - - -	B A ? ? ? ? ? ?	STU	0006h	- - D C	B A ? ? ? ? D C
ST	0003h	A - - -	A ? ? ? ? ? ? ?	ST	0003h	A - - -	A ? ? ? ? ? ? ?
STU	0005h	- - - B	A ? ? ? ? ? ? B	STU	0007h	- D C B	A ? ? ? ? D C B

【第37図】



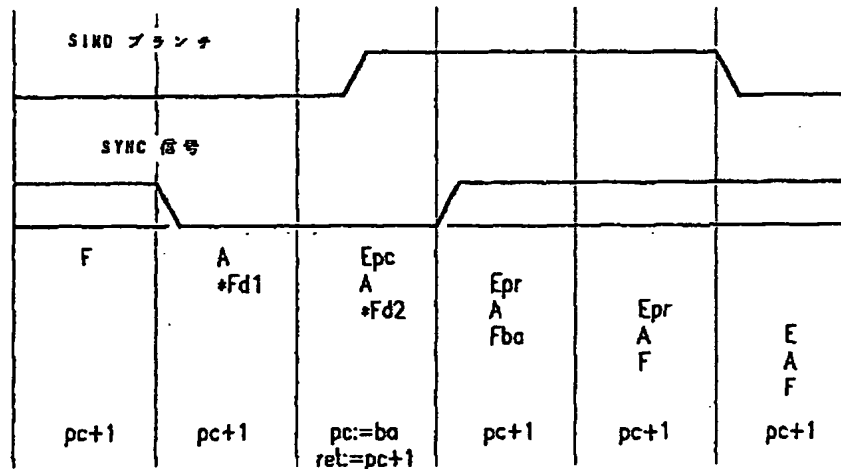
Fsq - ループのスタート・アドレス。

PC は正常にインクリメント。

Fer - エンド・アドレス。ループをリピートせよ。ループ・カウンタが1でない。PC にスタート・アドレスをロードせよ。

Fen - エンド・アドレス。ループ反復なし。ループ・カウンタが1である。PC は正常にインクリメント。

【第38図】



Epr - RET に PC + 1 をコピーせよ。

PC にブランチ・アドレスをロードせよ。

Epr - コールされた RET をプッシュせよ。

(Epr のどちらかがリターン・アドレスをプッシュ可能)

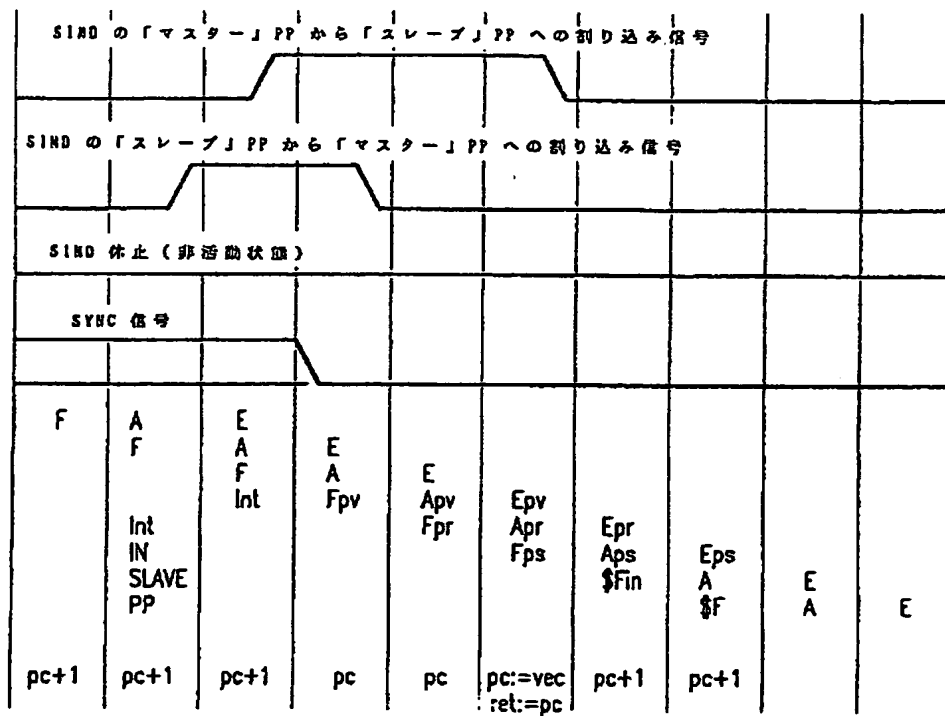
Fd1 - 記憶スロット1のQQフリップ

Fd2 - 記憶スロット2のQQフリップ

Fba - ブランチ・アドレスからのフリップQQ

F - リトリビュート

【第39図】



Int - 割り込み発生

Fpv - 戻り命令

(PC から RET へ、PC へのベクトル・フォーマット)

Apv - 割り込みベクトル・アドレスを計算せよ。

Fpr - PC を RET にコピーせよ。

割り込みベクトルを PC にフォーマットせよ。

Fps - 戻り命令。(RET をブロックせよ)

Apr - スタック・ブロック・アドレスを計算せよ。

Epr - RET をスタック上にブロックせよ。

Fps - 戻り命令。(SE をブロックせよ)

Aps - スタック・ブロック・アドレスを計算せよ。

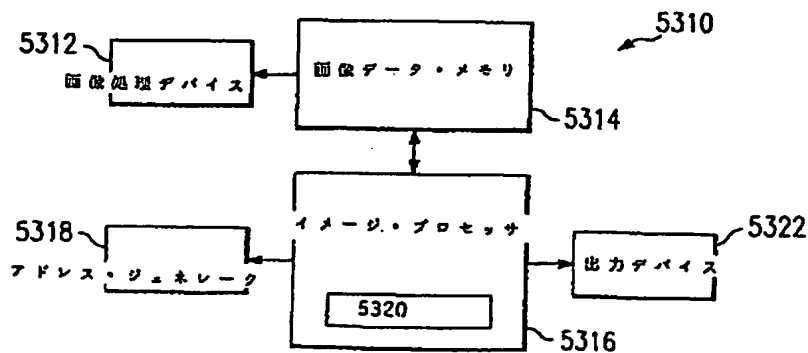
Eps - スタック上に SE をブロックせよ。

SE の S.I 及び CLO ビットをクリアせよ。

Fps - 割り込みルーチンの最初の命令。

1 - SE がブロックされるまで待機。割り込み及びルーチン・レベルはディスエンブル状態に置かれる。割り込みルーチンの最初の2つの命令はどちらも LSI であってはならない。

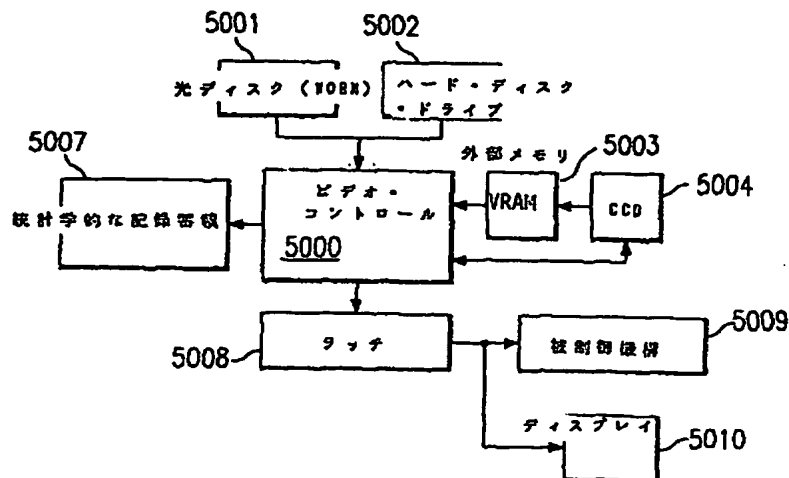
【第53図】



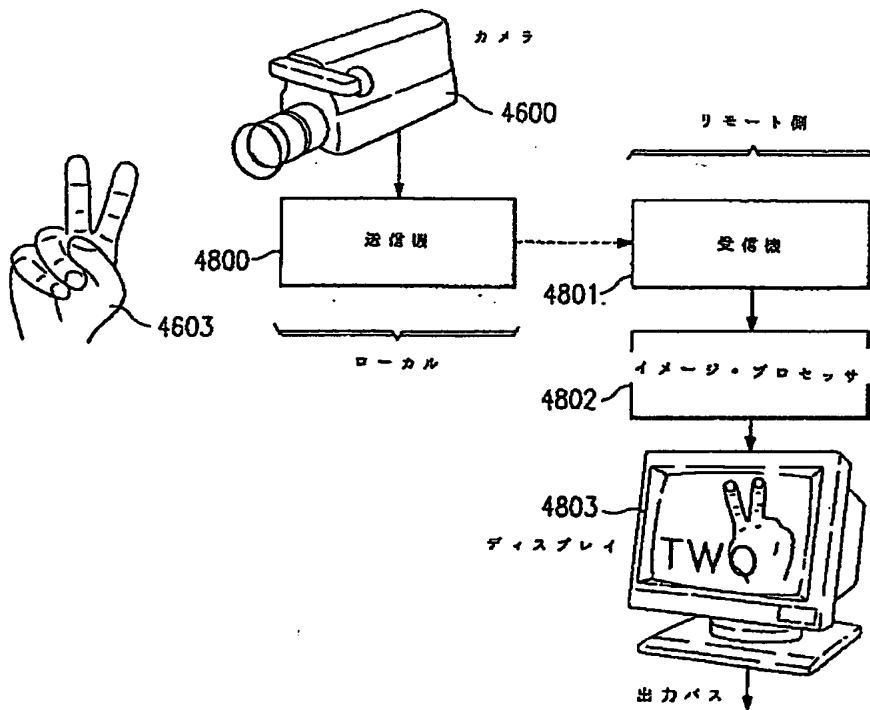
【第44図】

飽和加算		最 大		透明度	
ADDM	D0, D1, D2	SUBM	D0, D1, D2	CMPM	D0, D1
MRGM	D2, D3, D2	MRGM	D0, D1, D2	MRGM	D0, D2, D3
D0 = 89 23 CD 67		D0 = 89 23 CD 67		D0 = 89 23 CD 67	
+D1 = 01 AB 45 EF		-D1 = 01 AB 45 EF		(-)D1 = 23 23 23 23	
D2: = 8A CE 12 56		D2: = 88 67 88 67		(= 66 00 8A 44)	
MFLAGS: = ?? ?? ?? ?3		MFLAGS: = ?? ?? ?? ?5		MFLAGS: = ?? ?? ?? ?4	
D2 = 8A CE 12 56		D0 = 89 23 CD 67		D0 = 89 23 CD 67	
D3 = FF FF FF FF		D1 = 01 AB 45 EF		D2 = 87 65 43 21	
D2: = 8A CE FF FF		D2: = 89 AB CD EF		D3: = 89 65 CD 67	
色 膨 張		色 圧 縮		ナイフ・コピー	
LD	*AD, MFLAGS	CMPM	D0, D1, D2	LD	*AD, MFLAGS
MRGM	D0, D1, D2			MRGM	D0, D1, D1
		D0 = 89 23 CD 67			
		(-)D1 = 89 89 89 89			
		(= 00 89 44 CD)			
MFLAGS = XX XX XX X6		MFLAGS = ?? ?? ?? ?8		MFLAGS = XX XX XX XC	
D0 = 11 11 11 11				D0 = 89 23 CD 67	
D1 = 88 88 88 88				D1 = 87 65 43 21	
D2: = 11 88 88 11				D1: = 87 65 CD 67	

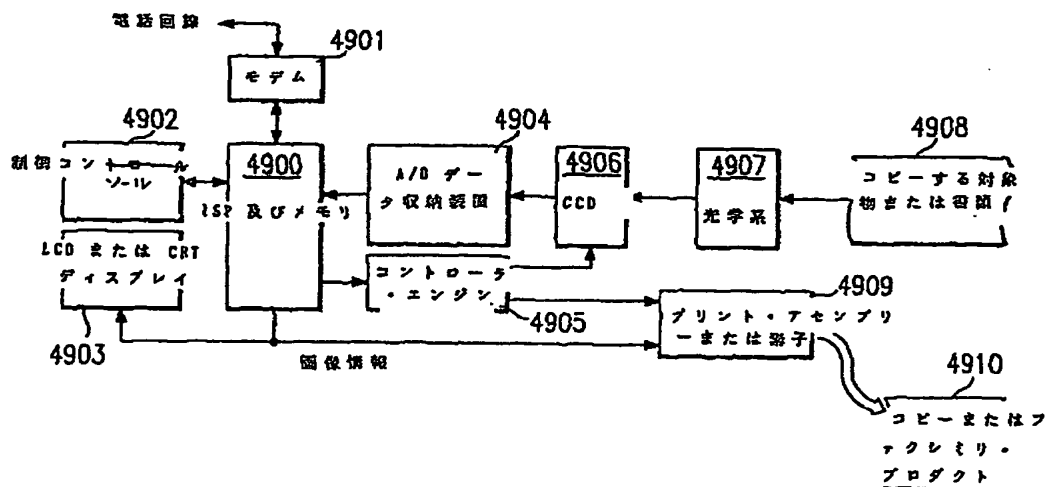
【第50図】



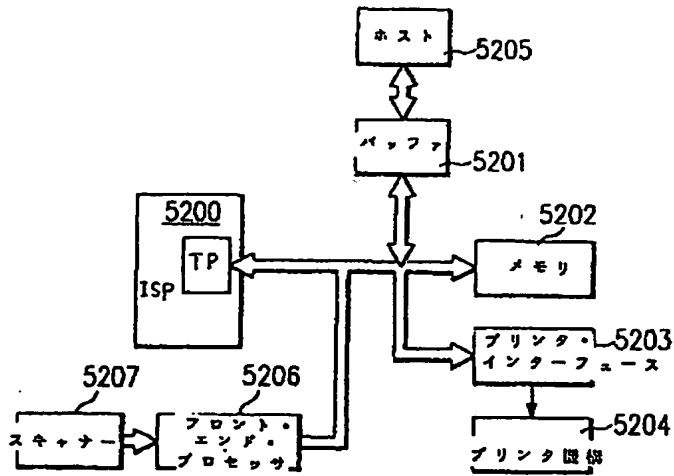
【第48図】



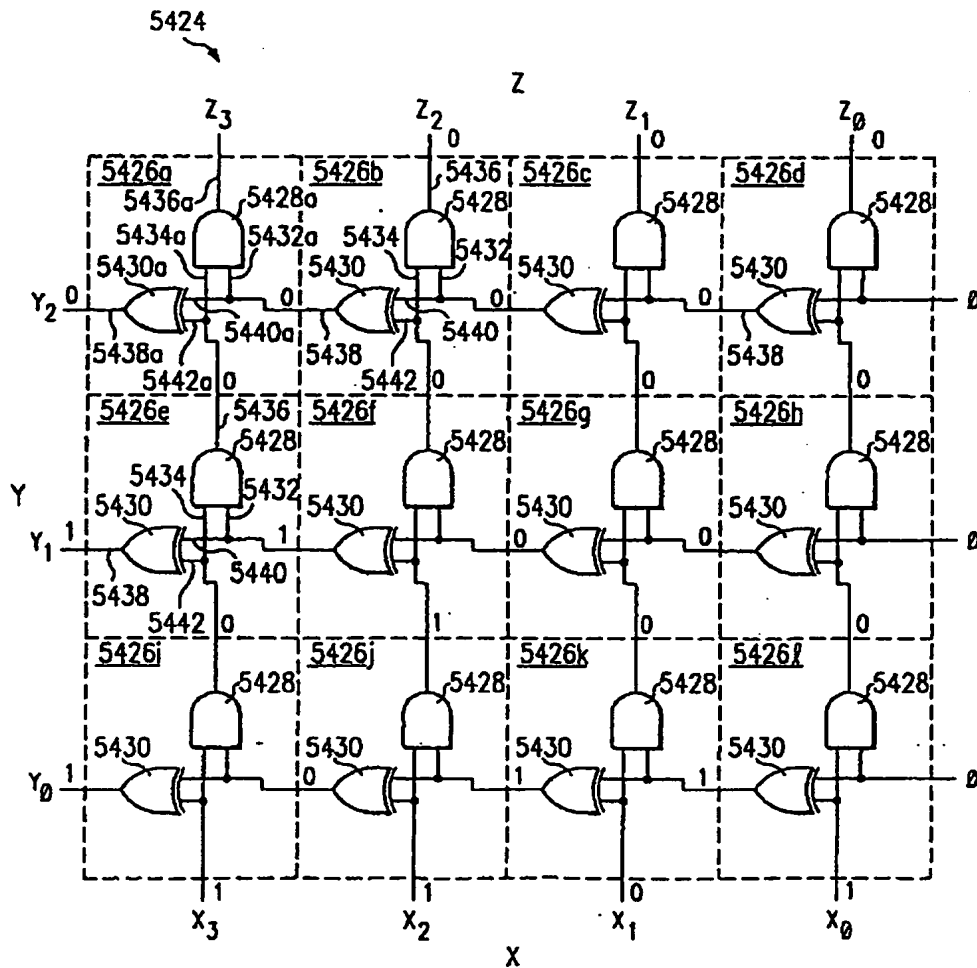
【第49図】



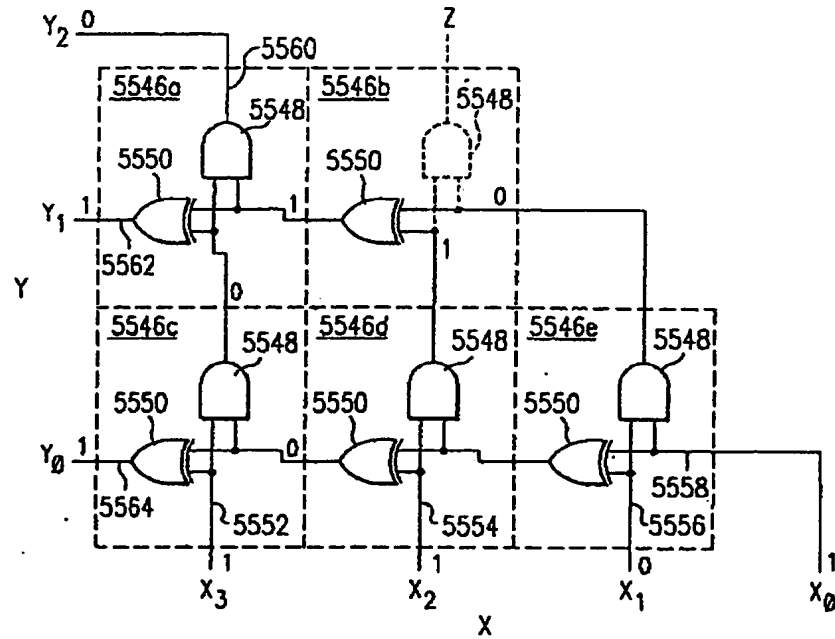
【第52図】



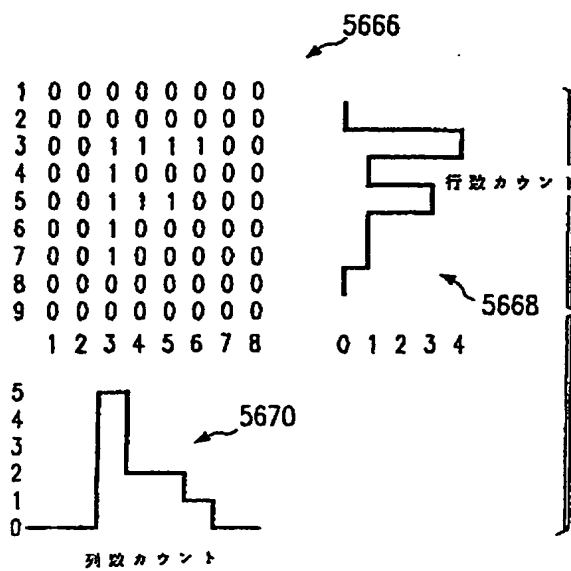
【第54図】



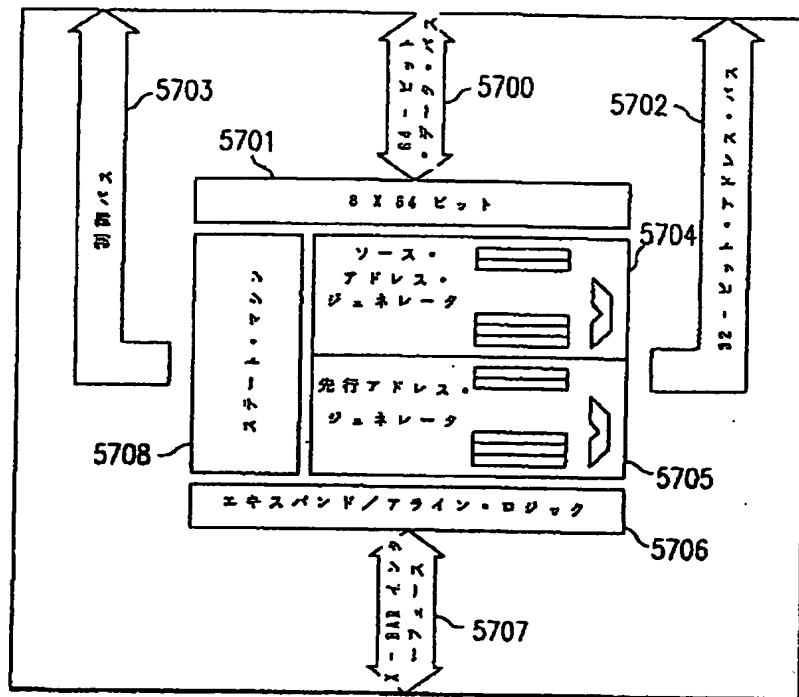
【第 5 5 図】



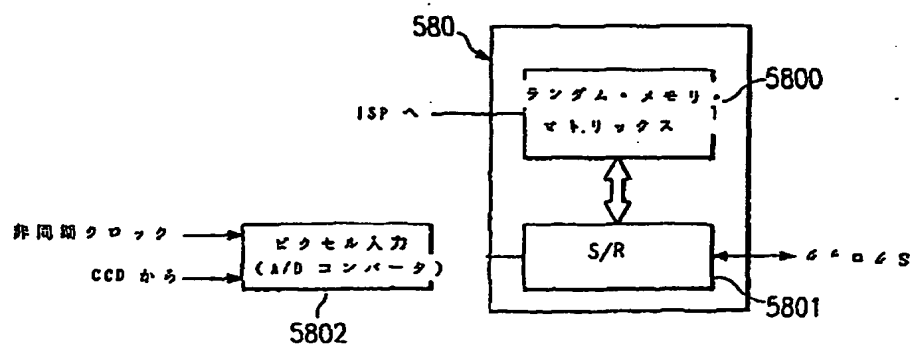
【第 5 6 図】



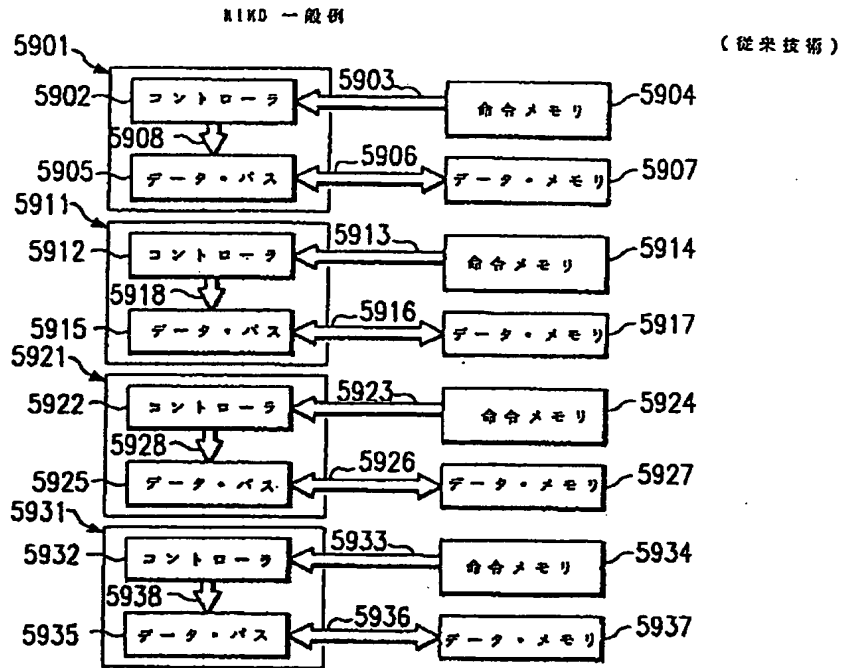
【第57図】



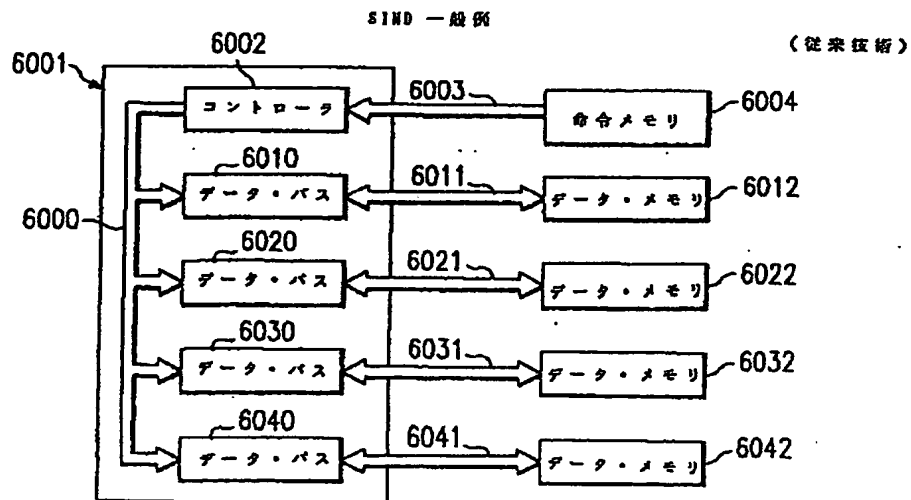
【第58図】



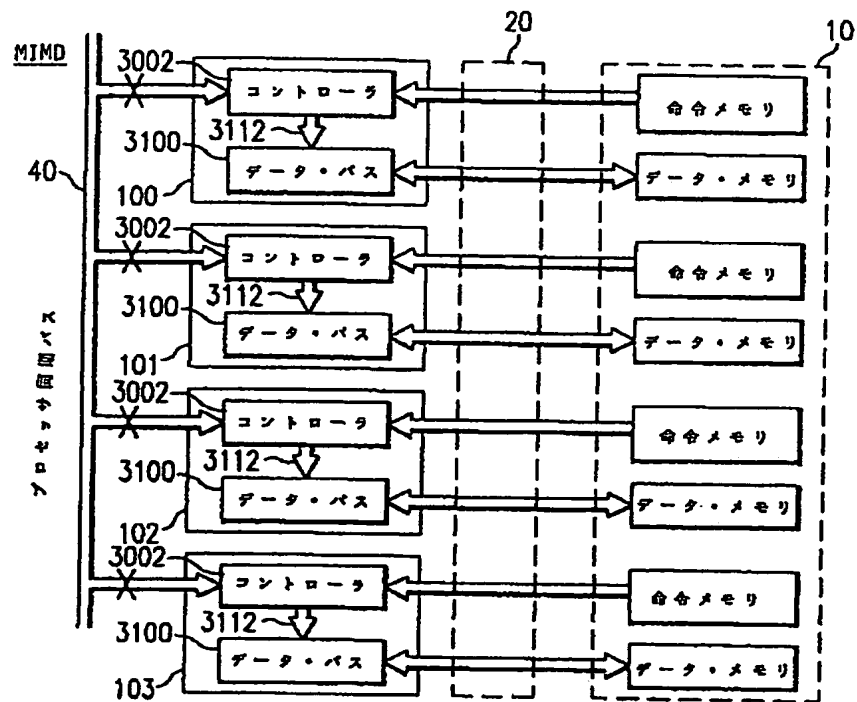
【第59図】



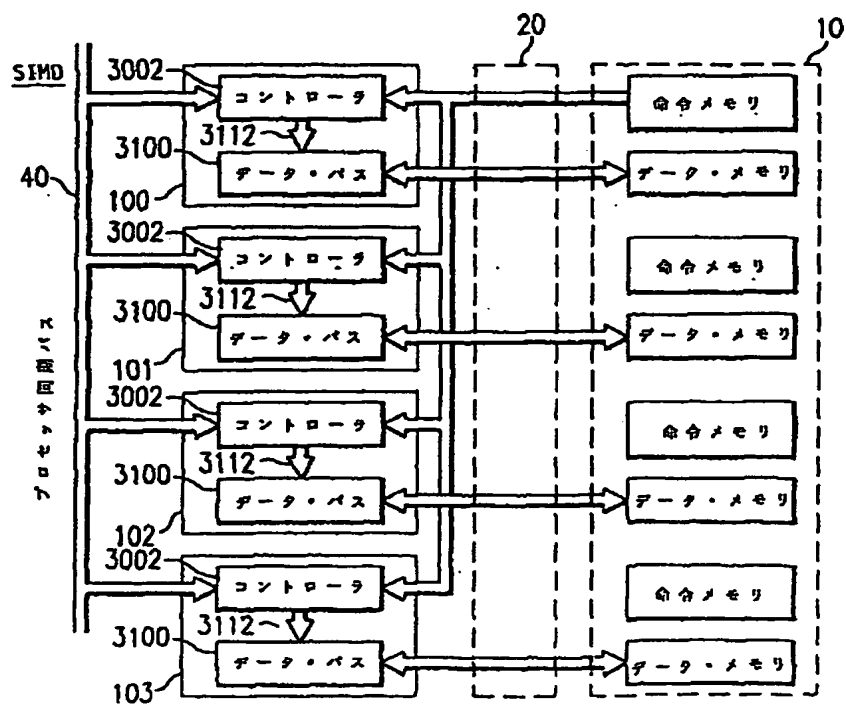
【第60図】



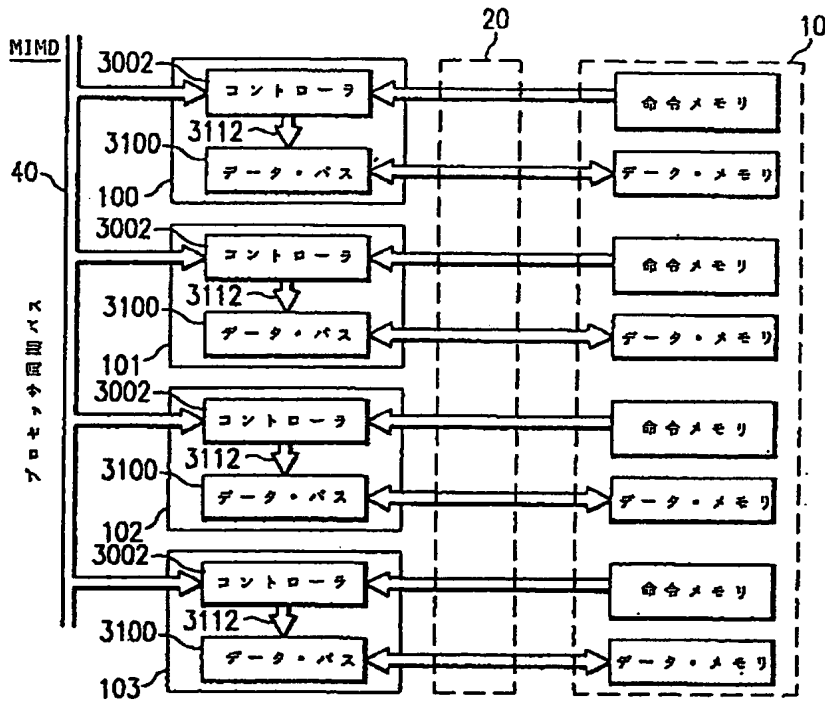
【第61図】



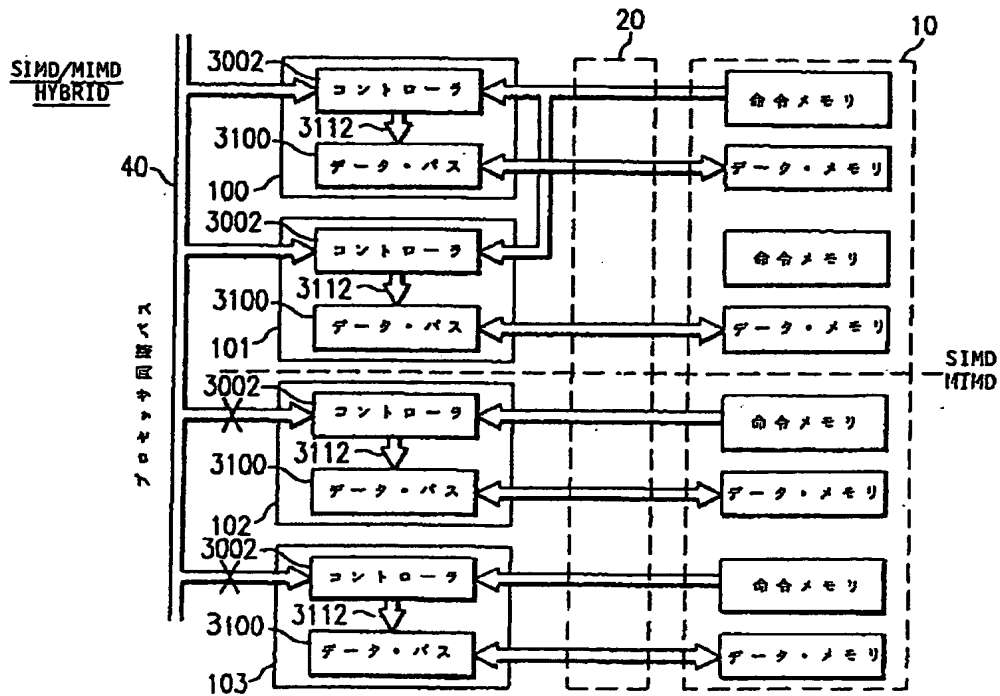
【第62図】



【第63図】



【第64図】



フロントページの続き

(72)発明者 カール エム グタッグ
 アメリカ合衆国 テキサス州 77459
 ミズーリー シティー サウス サンデ
 イー コート 4015
(72)発明者 ロバート ジェイ ゴーヴ
 アメリカ合衆国 テキサス州 75075
 ブラノ スカーボロー レーン 1405

(56)参考文献 特開 昭61-288260 (J P, A)
 特開 平 1 -232463 (J P, A)
 特開 昭64-105642 (J P, A)
 特開 昭62-259164 (J P, A)
 特開 平 1 -265355 (J P, A)

(58)調査した分野(Int. Cl.¹, DB名)

G06F 15/173

W P I